stdatamodels

Release 1.10.2.dev2+g4dfbff0.d20240410

STScI

Apr 10, 2024

CONTENTS:

CHAPTER

ONE

ASDFINFITS

stdatamodels.asdf_in_fits contains functions to help migrate code that uses the AsdfInFits format (ASDF data stored in a FITS file) from ASDF (which is dropping support for AsdfInFits) to stdatamodels.

1.1 Opening AsdfInFits files

AsdfInFits files can be opened with *stdatamodels.asdf_in_fits.open()*. This function aims to replace previous calls to `asdf.open()` or `asdf.AsdfInFits.open()` and can similarly be used in a with statement.

```
with stdatamodels.asdf_in_fits.open('some_file.fits') as af:
    # access the contents of `af` as a normal `asdf.AsdfFile`
    pass
```

It is recommended that a with statement is used to ensure the resulting asdf.AsdfFile and astropy.io.fits.HDUList are closed properly.

If a `with` statement cannot be used, *stdatamodels.asdf_in_fits.open()* can be used as a regular function (be sure to close the file when done).

```
af = stdatamodels.asdf_in_fits.open('some_file.fits')
# access the contents of `af` as a normal `asdf.AsdfFile`
af.close()
```

1.2 Writing AsdfInFits files

stdatamodels.asdf_in_fits.write() can be used to write ASDF data within a FITS file.

```
tree = {'sci': [1, 2, 3]}   # data to be stored in ASDF format
stdatamodels.asdf_in_fits.write('some_file.fits', tree)
```

This functions is meant to replace calls to `asdf.AsdfInFits.write_to()`. Please see the *stdatamodels.asdf_in_fits.write()* documentation for all available arguments.

stdatamodels.asdf_in_fits.write() supports references to array data within the `astropy.io.fits.HDUList`. This can be accomplished by first creating a `astropy.io.fits.HDUList` and populating it with data

```
from astropy.io import fits

hdulist = fits.HDUList()
```

(continues on next page)

(continued from previous page)

```
hdulist.append(fits.ImageHDU(np.arange(512, dtype=float), name='SCI'))
hdulist.append(fits.ImageHDU(np.arange(512, dtype=float), name='DQ'))
```

Then constructing a tree with references to the **same** data.

```
tree = {
    'model': {
        'sci': {
            'data': hdulist['SCI'].data,
        },
        'dq': {
            'data': hdulist['DQ'].data,
        }
    }
}
```

Finally providing the tree and hdulist to *stdatamodels.asdf_in_fits.write()*.

```
stdatamodels.asdf_in_fits.write('some_file.fits', tree, hdulist)
```

When read back with *stdatamodels.asdf_in_fits.open()* the data for `sci` and `dq` will be read from the HDUList instead of from the ASDF data embeded in the HDUList.

CHAPTER

TWO

JWST

2.1 Data Models

2.1.1 About models

The purpose of the data model is to abstract away the peculiarities of the underlying file format. The same data model may be used for data created from scratch in memory, or loaded from FITS or ASDF files or some future file format.

Hierarchy of models

There are different data model classes for different kinds of data.

One model instance, many arrays

Each model instance generally has many arrays that are associated with it. For example, the *ImageModel* class has the following arrays associated with it:

- *data*: The science data
- *dq*: The data quality array
- *err*: The error array

The shape of these arrays must be broadcast-compatible. If you try to assign an array to one of these members that is not broadcast-compatible with the data array, an exception is raised.

2.1.2 Working with models

Creating a data model from scratch

To create a new *ImageModel*, just call its constructor. To create a new model where all of the arrays will have default values, simply provide a shape as the first argument:

```
from stdatamodels.jwst.datamodels import ImageModel
with ImageModel((1024, 1024)) as im:
    ...
```

In this form, the memory for the arrays will not be allocated until the arrays are accessed. This is useful if, for example, you don't need a data quality array – the memory for such an array will not be consumed:

```
# Print out the data array.  It is allocated here on first access
# and defaults to being filled with zeros.
print(im.data)
```

If you already have data in a numpy array, you can also create a model using that array by passing it in as a data keyword argument:

```
data = np.empty((50, 50))
dq = np.empty((50, 50))
with ImageModel(data=data, dq=dq) as im:
    ...
```

Creating a data model from a file

The *jwst.datamodels.open* function is a convenient way to create a model from a file on disk. It may be passed any of the following:

- a path to a FITS file
- a path to an ASDF file
- a *astropy.io.fits.HDUList* object
- a readable file-like object

The file will be opened, and based on the nature of the data in the file, the correct data model class will be returned. For example, if the file contains 2-dimensional data, an *ImageModel* instance will be returned. You will generally want to instantiate a model using a *with* statement so that the file will be closed automatically when exiting the *with* block.

```
from stdatamodels.jwst import datamodels
with datamodels.open("myimage.fits") as im:
    assert isinstance(im, datamodels.ImageModel)
```

If you know the type of data stored in the file, or you want to ensure that what is being loaded is of a particular type, use the constructor of the desired concrete class. For example, if you want to ensure that the file being opened contains 2-dimensional image data:

```
from stdatamodels.jwst.datamodels import ImageModel
with ImageModel("myimage.fits") as im:
    # raises exception if myimage.fits is not an image file
    pass
```

This will raise an exception if the file contains data of the wrong shape.

Saving a data model to a file

Simply call the *save* method on the model instance. The format to save into will either be deduced from the filename (if provided) or the *format* keyword argument:

```
im.save("myimage.fits")
```

Note: Unlike `astropy.io.fits`, *save* always clobbers the output file.

Copying a model

To create a new model based on another model, simply use its *copy* method. This will perform a deep-copy: that is, no changes to the original model will propagate to the new model:

```
new_model = old_model.copy()
```

It is also possible to copy all of the known metadata from one model into a new one using the update method:

```
new_model.update(old_model)
```

History information

Models contain a list of history records, accessed through the *history* attribute. This is just an ordered list of strings – nothing more sophisticated.

To get to the history:

```
entries = model.history
for entry in entries:
  pass
```

To add an entry to the history, first create the entry by calling *stdatamodels.util.create_history_entry* and appending the entry to the model history:

```
import stdatamodels
entry = stdatamodels.util.create_history_entry("Processed through the frobulator step")
model.history.append(entry)
```

These history entries are stored in `HISTORY` keywords when saving to FITS format. As an option, history entries can contain a dictionary with a description of the software used. The dictionary must have the following keys:

> `name`: The name of the software `author`: The author or institution that produced the software `homepage`: A URI to the homepage of the software `version`: The version of the software

The calling sequence to create a history entry with the software description is:

```
entry =  stdatamodels.util.create_history_entry(description, software=software_dict)
```

where the second argument is the dictionary with the keywords mentioned.

Looking at the contents of a model

Use `model.info()` to look at the contents of a data model. It renders the underlying ASDF tree starting at the root or a specified `node`. The number of displayed rows is controlled by the `max_row` argument:

```
im.info()
root.tree (AsdfObject)
├─asdf_library (Software)
│  ├─author (str): Space Telescope Science Institute
│  ├─homepage (str): http://github.com/spacetelescope/asdf
│  ├─name (str): asdf
│  └─version (str): 2.5.2a1.dev12+g12aa460
├─history (dict)
```

(continues on next page)

(continued from previous page)

```
  └─extensions (list) ...
├─data (ndarray): shape=(2048, 2048), dtype=float32
├─dq (ndarray): shape=(2048, 2048), dtype=uint32
├─err (ndarray): shape=(2048, 2048), dtype=float32
├─meta (dict)
  ├─aperture (dict) ...
  ├─bunit_data (str): DN/s
  ├─bunit_err (str): DN/s
  ├─cal_step (dict) ...
  ├─calibration_software_revision (str): 3bfd782b
  ├─calibration_software_version (str): 0.14.3a1.dev133+g3bfd782b.d20200216
  ├─coordinates (dict) ...
  └─28 not shown
├─var_poisson (ndarray): shape=(2048, 2048), dtype=float32
├─var_rnoise (ndarray): shape=(2048, 2048), dtype=float32
└─extra_fits (dict) ...
Some nodes not shown.
```

Searching a model

`model.search()` can be used to search the ASDF tree by `key` or `value`:

```
im.search(key='filter')

root.tree (AsdfObject)
└─meta (dict)
├─instrument (dict)
  └─filter (str): F170LP
└─ref_file (dict)
  └─filteroffset (dict)
```

2.1.3 Converting from `astropy.io.fits`

This section describes how to port code that uses `astropy.io.fits` to use *jwst.datamodels*.

Opening a file

Instead of:

```
astropy.io.fits.open("myfile.fits")
```

use:

```
from stdatamodels.jwst.datamodels import ImageModel
with ImageModel("myfile.fits") as model:
    ...
```

In place of *ImageModel*, use the type of data one expects to find in the file. For example, if spectrographic data is expected, use *SpecModel*. If it doesn't matter (perhaps the application is only sorting FITS files into categories) use the base class *JwstDataModel*.

An alternative is to use:

```
from stdatamodels.jwst import datamodels
with datamodels.open("myfile.fits") as model:
    ...
```

The *datamodels.open()* method checks if the *DATAMODL* FITS keyword has been set, which records the DataModel that was used to create the file. If the keyword is not set, then *datamodels.open()* does its best to guess the best DataModel to use.

Accessing data

Data should be accessed through one of the pre-defined data members on the model (*data*, *dq*, *err*). There is no longer a need to hunt through the HDU list to find the data.

Instead of:

```
hdulist['SCI'].data
```

use:

```
model.data
```

Accessing keywords

The data model hides direct access to FITS header keywords. Instead, use the *Metadata* tree.

There is a convenience method, *find_fits_keyword* to find where a FITS keyword is used in the metadata tree:

```
>>> from stdatamodels.jwst.datamodels import JwstDataModel
>>> # First, create a model of the desired type
>>> model = JwstDataModel()
>>> model.find_fits_keyword('DATE-OBS')
[u'meta.observation.date']
```

This information shows that instead of:

```
print(hdulist[0].header['DATE-OBS'])
```

use:

```
print(model.meta.observation.date)
```

Extra FITS keywords

When loading arbitrary FITS files, there may be keywords that are not listed in the schema for that data model. These "extra" FITS keywords are put under the model in the *_extra_fits* namespace.

Under the *_extra_fits* namespace is a section for each header data unit, and under those are the extra FITS keywords. For example, if the FITS file contains a keyword *FOO* in the primary header, its value can be obtained using:

```
model._extra_fits.PRIMARY.FOO
```

This feature is useful to retain any extra keywords from input files to output products.

To get a list of everything in *_extra_fits*:

```
model._extra_fits._instance
```

returns a dictionary of of the instance at the model._extra_fits node.

_instance can be used at any node in the tree to return a dictionary of rest of the tree structure at that node.

Environment Variables

There are a number of environment variables that affect how models are read.

PASS_INVALID_VALUES
: Used by *~jwst.datamodels.JwstDataModel* when instantiating a model from a file. If `True`, values that do not validate the schema will still be added to the metadata. If `False`, they will be set to `None`. Default is `False`.

STRICT_VALIDATION
: Used by *~jwst.datamodels.JwstDataModel* when instantiating a model from a file. If `True`, schema validation errors will generate an exception. If `False`, they will generate a warning. Default is `False`.

SKIP_FITS_UPDATE
: DEPRECATED: In the future the FITS header will always be used. Used by *~jwst.datamodels.JwstDataModel* when instantiating a model from a FITS file. When `False`, models opened from FITS files will proceed and load the FITS header values into the model. When `True` and the FITS file has an ASDF extension, the loading/validation of the FITS header will be skipped, loading the model only from the ASDF extension. If not defined, the instantiation routines will determine whether the loading/validation of the FITS header can be skipped or not.

DMODEL_ALLOWED_MEMORY
: Implemented by the utility function *jwst.datamodels.util.check_memory_allocation* and used by *~jwst.outlier_detection.OutlierDetectionStep* and *~jwst.resample.ResampleStep*. When defined, determines how much of currently available memory should be used to instantiated an output resampled image. If not defined, no check is made.

  Examples would be: `1.0` would allow all available memory to be used. `0.5` would allow only half the available memory to be used.

For flag or boolean variables, any value in `('true', 't', 'yes', 'y')` or a non-zero number, will evaluate as `True`. Any value in `('false', 'f', 'no', 'n', '0')` will evaluate as `False`. The values are case-insensitive.

All of the environment variables have equivalent function arguments in the API for the relevant code. The environment variables are used only if explicit values had not been used in a script. In other words, values in code override environment variables.

2.1.4 Data model attributes

The purpose of the data model is to abstract away the peculiarities of the underlying file format. The same data model may be used for data created from scratch in memory, loaded from FITS or ASDF files, or from some other future format.

2.1.5 Calling sequences of models

List of current models

The current models are as follows:

'ABVegaOffsetModel', 'AmiLgModel', 'FgsImgApcorrModel', 'MirImgApcorrModel', 'NrcImgApcorrModel', 'NisImgApcorrModel', 'MirLrsApcorrModel', 'MirMrsApcorrModel', 'NrcWfssApcorrModel', 'NisWfssApcorrModel', 'NrsMosApcorrModel', 'NrsFsApcorrModel', 'NrsIfuApcorrModel', 'AsnModel', 'BarshadowModel', 'CameraModel', 'CollimatorModel', 'CombinedSpecModel', 'ContrastModel', 'CubeModel', 'DarkModel', 'DarkMIRIModel', 'DisperserModel', 'DistortionModel', 'DistortionMRSModel', 'DrizParsModel', 'Extract1dImageModel', 'Extract1dIFUModel', 'FilteroffsetModel', 'FlatModel', 'NirspecFlatModel', 'NirspecQuadFlatModel', 'FOREModel', 'FPAModel', 'FringeModel', 'GainModel', 'GLS_RampFitModel', 'GuiderRawModel', 'GuiderCalModel', 'IFUCubeModel', 'NirspecIFUCubeParsModel', 'MiriIFUCubeParsModel', 'IFUFOREModel', 'IFUImageModel', 'IFUPostModel', 'IFUSlicerModel', 'ImageModel', 'IPCModel', 'IRS2Model', 'LastFrameModel', 'Level1bModel', 'LinearityModel', 'MaskModel', 'ModelContainer', 'MSAModel', 'MultiCombinedSpecModel', 'MultiExposureModel', 'MultiExtract1dImageModel', 'MultiSlitModel', 'MultiSpecModel', 'NIRCAMGrismModel', 'NIRISSGrismModel', 'OTEModel', 'OutlierParsModel', 'PathlossModel', 'PersistenceSatModel', 'PixelAreaModel', 'NirspecSlitAreaModel', 'NirspecMosAreaModel', 'NirspecIfuAreaModel', 'FgsImgPhotomModel', 'MirImgPhotomModel', 'MirLrsPhotomModel', 'MirMrsPhotomModel', 'NrcImgPhotomModel', 'NrcWfssPhotomModel', 'NisImgPhotomModel', 'NisSossPhotomModel', 'NisWfssPhotomModel', 'NrsFsPhotomModel', 'NrsMosPhotomModel', 'PsfMaskModel', 'QuadModel', 'RampModel', 'RampFitOutputModel', 'ReadnoiseModel', 'ReferenceFileModel', 'ReferenceCubeModel', 'ReferenceImageModel', 'ReferenceQuadModel', 'RegionsModel', 'ResetModel', 'ResolutionModel', 'MiriResolutionModel', 'RSCDModel', 'SaturationModel', 'SlitDataModel', 'SlitModel', 'SpecModel', 'SegmentationMapModel', 'SourceModelContainer', 'SpecKernelModel', 'SpecProfileModel', 'SpecProfileSingleModel', 'SpecTraceModel', 'SpecTraceSingleModel', 'SpecwcsModel', 'StrayLightModel', 'SuperBiasModel', 'ThroughputModel', 'TrapDensityModel', 'TrapParsModel', 'TrapsFilledModel', 'TsoPhotModel', 'WavelengthrangeModel', 'WaveCorrModel', 'WaveMapModel', 'WaveMapSingleModel', 'WfssBkgModel'

Commonly used attributes

Here are a few model attributes that are used by some of the pipeline steps.

For uncalibrated data *_uncal.fits*. Getting the number of integrations and the number of groups from the first and second axes assumes that the input data array is 4-D data. Pixel coordinates in the data extensions are 1-indexed as in FORTRAN and FITS headers, not 0-indexed as in Python.

- `input_model.data.shape[0]`: number of integrations
- `input_model.data.shape[1]`: number of groups
- `input_model.meta.exposure.nframes`: number of frames per group
- **`input_model.meta.exposure.groupgap`: number of frames dropped** between groups
- `input_model.meta.subarray.xstart`: starting pixel in X (1-based)
- `input_model.meta.subarray.ystart`: starting pixel in Y (1-based)
- `input_model.meta.subarray.xsize`: number of columns
- `input_model.meta.subarray.ysize`: number of rows

The *data*, *err*, *dq*, etc., attributes of most models are assumed to be numpy.ndarray arrays, or at least objects that have some of the attributes of these arrays. numpy is used explicitly to create these arrays in some cases (e.g. when a default value is needed). The *data* and *err* arrays are a floating point type, and the data quality arrays are an integer type.

Some of the step code makes assumptions about image array sizes. For example, full-frame MIRI data have 1032 columns and 1024 rows, and all other detectors have 2048 columns and rows; anything smaller must be a subarray. Also, full-frame MIRI data are assumed to have four columns of reference pixels on the left and right sides (the reference output array is stored in a separate image extension). Full-frame data for all other instruments have four columns or rows of reference pixels on each edge of the image.

JwstDataModel Base Class

class jwst.datamodels.**JwstDataModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If

'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

`property crds_observatory`

Get CRDS observatory code for this model.

Returns

str

`get_crds_parameters()`

Get parameters used by CRDS to select references for this model.

Returns

dict

`on_init(init)`

Hook invoked by the base class before returning a newly created model instance.

`on_save(init)`

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

`schema_url = 'http://stsci.edu/schemas/jwst_datamodel/core.schema'`

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

2.1.6 Metadata

Metadata information associated with a data model is accessed through its *meta* member. For example, to access the date that an observation was made:

```
print(model.meta.observation.date)
```

Metadata values are automatically type-checked against the schema when they are set. Therefore, setting a keyword which expects a number to a string will raise an exception:

```
>>> from stdatamodels.jwst.datamodels import ImageModel
>>> model = ImageModel()
>>> model.meta.target.ra = "foo"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "site-packages/jwst.datamodels/schema.py", line 672, in __setattr__
    object.__setattr__(self, attr, val)
  File "site-packages/jwst.datamodels/schema.py", line 490, in __set__
    val = self.to_basic_type(val)
  File "site-packages/jwst.datamodels/schema.py", line 422, in to_basic_type
    raise ValueError(e.message)
ValueError: 'foo' is not of type u'number'
```

The set of available metadata elements is defined in a YAML Schema that ships with *jwst.datamodels*.

There is also a utility method for finding elements in the metadata schema. *search_schema* will search the schema for the given substring in metadata names as well as their documentation. The search is case-insensitive:

```
>>> from stdatamodels.jwst.datamodels import ImageModel
>>> # Create a model of the desired type
>>> model = ImageModel()
>>> # Call `search_schema` on it to find possibly related elements.
>>> model.search_schema('target')
meta.target

meta.target.catalog_name

meta.target.category

meta.target.dec

meta.target.dec_uncertainty

meta.target.description

meta.target.proper_motion_dec

meta.target.proper_motion_epoch

meta.target.proper_motion_ra

meta.target.proposer_dec

meta.target.proposer_name

meta.target.proposer_ra

meta.target.ra

meta.target.ra_uncertainty

meta.target.source_type
```

(continues on next page)

(continued from previous page)

```
meta.target.source_type_apt

meta.target.type

meta.visit.internal_target
```

An alternative method to get and set metadata values is to use a dot-separated name as a dictionary lookup. This is useful for databases, such as CRDS, where the path to the metadata element is most conveniently stored as a string. The following two lines are equivalent:

```
print(model['meta.observation.date'])
print(model.meta.observation.date)
```

2.1.7 Working with lists

Unlike ordinary Python lists, lists in the schema may be restricted to only accept a certain set of values. Items may be added to lists in two ways: by passing a dictionary containing the desired key/value pairs for the object, or using the lists special method *item* to create a metadata object and then assigning that to the list.

For example, suppose the metadata element *meta.transformations* is a list of transformation objects, each of which has a *type* (string) and a *coeff* (number) member. We can assign elements to the list in the following equivalent ways:

```
.. doctest-skip::
```

```
>>> trans = model.meta.transformations.item()
>>> trans.type = 'SIN'
>>> trans.coeff = 42.0
>>> model.meta.transformations.append(trans)
>>> model.meta.transformations.append({'type': 'SIN', 'coeff': 42.0})
```

When accessing the items of the list, the result is a normal metadata object where the attributes are type-checked:

```
.. doctest-skip::
```

```
>>> trans = model.meta.transformations[0]
>>> print(trans)
<jwst.datamodels.schema.Transformations object at 0x123a810>
>>> print(trans.type)
SIN
>>> trans.type = 42.0
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "site-packages/jwst.datamodels/schema.py", line 672, in __setattr__
    object.__setattr__(self, attr, val)
  File "site-packages/jwst.datamodels/schema.py", line 490, in __set__
    val = self.to_basic_type(val)
  File "site-packages/jwst.datamodels/schema.py", line 422, in to_basic_type
    raise ValueError(e.message)
ValueError: 42.0 is not of type u'string'
```

2.1.8 JSON Schema

The *jwst.datamodels* library defines its metadata using Draft 4 of the JSON Schema specification, but jwst.datamodels uses YAML for the syntax. A good resource for learning about JSON schema is the book Understanding JSON Schema. The mapping from Javascript to Python concepts (such as Javascript "array" == Python "list") is added where applicable.

In addition to the standard JSON Schema keywords, `jwst.datamodels` also supports the following additional keywords.

Arrays

The following keywords have to do with validating n-dimensional arrays:

- `ndim`: The number of dimensions of the array.
- `max_ndim`: The maximum number of dimensions of the array.
- `datatype`: For defining an array, `datatype` should be a string. For defining a table, it should be a list.
- **array**: `datatype` should be one of the following strings, representing fixed-length datatypes:

  bool8, int8, int16, int32, int64, uint8, uint16, uint32, uint64, float16, float32, float64, float128, complex64, complex128, complex256

Or, for fixed-length strings, an array `[ascii, XX]` where `XX` is the maximum length of the string.

(Datatypes whose size depend on the platform are not supported since this would make files less portable).

- **table**: `datatype` should be a list of dictionaries. Each element in the list defines a column and has the following keys:
  - `datatype`: A string to select the type of the column. This is the same as the `datatype` for an array (as described above).
  - `name` (optional): An optional name for the column.
  - `shape` (optional): The shape of the data in the column. May be either an integer (for a single-dimensional shape), or a list of integers.

FITS-specific Schema Attributes

jwst.datamodels also adds some new keys to the schema language in order to handle reading and writing FITS files. These attributes all have the prefix `fits_`.

- `fits_keyword`: Specifies the FITS keyword to store the value in. Must be a string with a maximum length of 8 characters.
- `fits_hdu`: Specifies the FITS HDU to store the value in. May be a number (to specify the nth HDU) or a name (to specify the extension with the given `EXTNAME`). By default this is set to 0, and therefore refers to the primary HDU.

2.1.9 Creating a new model

This tutorial describes the steps necessary to define a new model type using *jwst.datamodels*.

For further reading and details, see the reference materials in *Metadata*.

In this tutorial, we'll go through the process of creating a new type of model for a file format used for storing the bad pixel mask for JWST's MIRI instrument. This file format has a 2D array containing a bit field for each of the pixels, and a table describing what each of the bits in the array means.

Note: While an attempt is made to present a real-world example here, it may not reflect the actual final format of this file type, which is still subject to change at the time of this writing.

This example will be built as a third-party Python package, i.e. not part of *jwst.datamodels* itself. Doing so adds a few extra wrinkles to the process, and it's most helpful to show what those wrinkles are. To skip ahead and just see the example in its entirety, see the `examples/custom_model` directory within the *jwst.datamodels* source tree.

Directory layout

The bare minimum directory layout for a Python package that creates a custom model is as below:

```
.
|-- lib
|   |--- __init__.py
|   |--- bad_pixel_mask.py
|   |--- schemas
|   |--- bad_pixel_mask.schema.yaml
|   |--- tests
|       |--- __init__.py
|       |--- test_bad_pixel_mask.py
|       |--- data
|       |--- bad_pixel_mask.fits
|--- setup.py
```

The main pieces are the new schema in `bad_pixel_mask.schema.yaml`, the custom model class in `bad_pixel_mask.py`, a `setup.py` file to install the package, and some unit tests and associated data. Normally, you would also have some code that *uses* the custom model included in the package, but that isn't included in this minimal example.

The schema file

Let's start with the schema file, `bad_pixel_mask.schema.yaml`. There are a few things it needs to do:

1) It should contain all of the core metadata from the core schema that ships with *jwst.datamodels*. In JSON Schema parlance, this schema "extends" the core schema. In object-oriented programming terminology, this could be said that our schema "inherits from" the core schema. It's all the same thing.

2) Define the pixel array containing the information about each of the bad pixels. This will be an integer for each pixel where each bit is ascribed a particular meaning.

3) Define a table describing what each of the bit fields in the pixel array means. This will have three columns: one for the bit field's number (a power of 2), one for a name token to identify it, and one with a human-readable description.

At the top level, every JSON schema must be a mapping (dictionary) of type "object", and should include the core schema:

```
allOf:
  - $ref: "http://jwst.stsci.edu/schemas/core.schema.yaml"
  - type: object
    properties:
      ...
```

There's a lot going on in this one item. `$ref` declares the schema fragment that we want to include (the "base class" schema). Here, the `$ref` mapping causes the system to go out and fetch the content at the given URL, and then replace the mapping with that content.

The `$ref` URL can be a relative URL, in which case it is relative to the schema file where `$ref` is used. In our case, however, it's an absolute URL. Before you visit that URL to see what's there, I'll save you the trouble: there is nothing at that HTTP address. The host `jwst.stsci.edu` is recognized as a "special" address by the system that causes the schema to be looked up alongside installed Python code. For example, to refer to a (hypothetical) `my_instrument` schema that ships with a Python package called `astroboy`, use the following URL:

```
http://jwst.stsci.edu/schemas/astroboy/my_instrument.schema.yaml
```

The "package" portion may be omitted to refer to schemas in the *jwst.datamodels* core, which is how we arrive at the URL we're using here:

```
http://jwst.stsci.edu/schemas/core.schema.yaml
```

Note: At some time in the future, we will actually be hosting schemas at a URL similar to the one above. This will allow schemas to be shared with tools built in languages other than Python. Until we have that hosting established, this works quite well and does not require any coordination among Python packages that define new models. Keep an eye out if you use this feature, though – the precise URL used may change.

The next part of the file describes the array data, that is, things that are Numpy arrays on the Python side and images or tables on the FITS side.

First, we describe the main `"dq"` array. It's declared to be 2-dimensional, and each element is an unsigned 32-bit integer:

```
properties:
  dq:
    title: Bad pixel mask
    fits_hdu: DQ
    default: 0
    ndim: 2
    datatype: uint16
```

The next entry describes a table that will store the mapping between bit fields and their meanings. This table has four columns:

- `BIT`: The value of the bit field (a power of 2)
- `VALUE`: The value resulting when raising 2 to the BIT power
- `NAME`: The name used to refer to the bit field
- `DESCRIPTION`: A longer, human-readable description of the bit field

```
dq_def:
  title: DQ flag definitions
  fits_hdu: DQ_DEF
  dtype:
    - name: BIT
      datatype: uint32
    - name: VALUE
      datatype: uint32
    - name: NAME
      datatype: [ascii, 40]
    - name: DESCRIPTION
      datatype: [ascii, 80]
```

And finally, we add a metadata element that is specific to this format. To avoid recomputing it repeatedly, we'd like to store a sum of all of the "bad" (i.e. non-zero) pixels stored in the bad pixel mask array. In the model, we want to refer to this value as `model.meta.bad_pixel_count`. In the FITS file, lets store this in the primary header in a keyword named `BPCOUNT`:

```
meta:
  properties:
    bad_pixel_count:
      type: integer
      title: Total count of all bad pixels
      fits_keyword: BPCOUNT
```

That's all there is to the schema file, and that's the hardest part.

The model class

Now, let's see how this schema is tied in with a new Python class for the model.

First, we need to import the *JwstDataModel* class, which is the base class for all models:

```
from stdatamodels.jwst.datamodels import JwstDataModel
```

Then we create a new Python class that inherits from *JwstDataModel*, and set its *schema_url* class member to point to the schema that we just defined above:

```
class MiriBadPixelMaskModel(JwstDataModel):
    schema_url = "bad_pixel_mask.schema.yaml"
```

Here, the *schema_url* has all of the "magical" URL abilities described above when we used the `$ref` feature. However, here we are using a relative URL. In this case, it is relative to the file in which this class is defined, with a small twist to avoid intermingling Python code and schema files: It looks for the given file in a directory called `schemas` inside the directory containing the Python module in which the class is defined.

As an alternative, we could just as easily have said that we want to use the `image` schema from the core without defining any extra elements, by setting *schema_url* to:

```
schema_url = "http://jwst.stsci.edu/schemas/image.schema.yaml"
```

Note: At this point you may be wondering why both the schema and the class have to inherit from base classes. Certainly, it would have been more convenient to have the inheritance on the Python side automatically create the

inheritance on the schema side (or vice versa). The reason we can't is that the schema files are designed to be language-agnostic: it is possible to use them from an entirely different implementation of the *jwst.datamodels* framework possibly even written in a language other than Python. So the schemas need to "stand alone" from the Python classes. It's certainly possible to have the schema inherit from one thing and the Python class inherit from another, and the *jwst.datamodels* framework won't and can't really complain, but doing that is only going to lead to confusion, so just don't do it.

Within this class, we'll define a constructor. All model constructors must take the highly polymorphic `init` value as the first argument. This can be a file, another model, or all kinds of other things. See the docstring of *jwst.datamodels.JwstDataModel.__init__* for more information. But we're going to let the base class handle that anyway.

The rest of the arguments are up to you, but generally it's handy to add a couple of keyword arguments so the user can data arrays when creating a model from scratch. If you don't need to do that, then technically writing a new constructor for the model is optional:

```
def __init__(self, init=None, dq=None, dq_def=None, **kwargs):
    """
    A data model to represent MIRI bad pixel masks.

    Parameters
    ----------
    init : any
        Any of the initializers supported by `~jwst.datamodels.JwstDataModel`.

    dq : numpy array
        The data quality array.

    dq_def : numpy array
        The data quality definitions table.
    """
    super(MiriBadPixelMaskModel, self).__init__(init=init, **kwargs)

    if dq is not None:
        self.dq = dq

    if dq_def is not None:
        self.dq_def = dq_def
```

The `super..` line is just the standard Python way of calling the constructor of the base class. The rest of the constructor sets the arrays on the object if any were provided.

The other methods of your class may provide additional conveniences on top of the underlying file format. This is completely optional and if your file format is supported well enough by the underlying schema alone, it may not be necessary to define any extra methods.

In the case of our example, it would be nice to have a function that, given the name of a bit field, would return a new array that is *True* wherever that bit field is true in the main mask array. Since the order and content of the bit fields are defined in the *dq_def* table, the function should use it in order to do this work:

```
def get_mask_for_field(self, name):
    """
    Returns an array that is `True` everywhere a given bitfield is
    True in the mask.
```

(continues on next page)

(continued from previous page)

```
    Parameters
    ----------
    name : str
        The name of the bit field to retrieve

    Returns
    -------
    array : boolean numpy array
        `True` everywhere the requested bitfield is `True`.  This
        is the same shape as the mask array.  This array is a copy
        and changes to it will not affect the underlying model.
    """
    # Find the field value that corresponds to the given name
    field_value = None
    for value, field_name, title in self.dq_def:
        if field_name == name:
            field_value = value
            break
    if field_value is None:
        raise ValueError("Field name {0} not found".format(name))

    # Create an array that is `True` only for the requested
    # bit field
    return self.dq & field_value
```

One thing to note here: this array is semantically a "copy" of the underlying data. Most Numpy arrays in the model framework are mutable, and we expect that changing their values will update the model itself, and be saved out by subsequent saves to disk. Since the array we are returning here has no connection back to the model's main data array (`mask`), it's helpful to remind the user of that in the docstring, and not present it as a member or property, but as a getter function.

Note: Since handling bit fields like this is such a commonly useful thing, it's possible that this functionality will become a part of *jwst.datamodels* itself in the future. However, this still stands as a good example of something someone may want to do in a custom model class.

Lastly, remember the `meta.bad_pixel_count` element we defined above? We need some way to make sure that whenever the file is written out that it has the correct value. The model may have been loaded and modified. For this, *JwstDataModel* has the *on_save* method hook, which may be overridden by the subclass to add anything that should happen just before saving:

```
def on_save(self, path):
    super(MiriBadPixelMaskModel, self).on_save(path)

    self.meta.bad_pixel_count = np.sum(self.mask != 0)
```

Note that here, like in the constructor, it is important to "chain up" to the base class so that any things that the base class wants to do right before saving also happen.

The *setup.py* script

Writing a `setup.py` script is beyond the scope of this tutorial but it's worth noting one thing. Since the schema files are not Python files, they are not automatically picked up by `setuptools`, and must be included in the `package_data` option. A complete, yet minimal, `setup.py` is presented below:

```
#!/usr/bin/env python

from setuptools import setup

setup(
    name='custom_model',
    description='Custom model example for jwst.datamodels',
    packages=['custom_model', 'custom_model.tests'],
    package_dir={'custom_model': 'lib'},
    package_data={'custom_model': ['schemas/*.schema.yaml'],
                  'custom_model.tests' : ['data/*.fits']}
    )
```

Using the new model

The new model can now be used. For example, to get the locations of all of the "hot" pixels:

```
from custom_model.bad_pixel_mask import MiriBadPixelMaskModel

with MiriBadPixelMaskModel("bad_pixel_mask.fits") as dm:
    hot_pixels = dm.get_mask_for_field('HOT')
```

A table-based model

In addition to n-dimensional data arrays, models can also contain tabular data. For example, the photometric correction reference file used in the JWST calibration pipeline consists of a table with several columns. The schema file for one of these models looks like this:

```
title: NIRISS SOSS photometric flux conversion data model
allOf:
  - $ref: "referencefile.schema.yaml"
  - $ref: "keyword_exptype.schema.yaml"
  - $ref: "keyword_pexptype.schema.yaml"
  - $ref: "keyword_pixelarea.schema.yaml"
  - type: object
    properties:
      phot_table:
        title: Photometric flux conversion factors table
        fits_hdu: PHOTOM
        datatype:
          - name: filter
            datatype: [ascii, 12]
          - name: pupil
            datatype: [ascii, 15]
          - name: order
```

(continues on next page)

(continued from previous page)

```
          datatype: int16
        - name: photmj
          datatype: float32
        - name: uncertainty
          datatype: float32
        - name: nelem
          datatype: int16
        - name: wavelength
          datatype: float32
          ndim: 1
        - name: relresponse
          datatype: float32
          ndim: 1
        - name: reluncertainty
          datatype: float32
          ndim: 1
```

In this particular table the first 6 columns contain scalar entries of types string, float, and integer. The entries in the final 3 columns, on the other hand, contain 1-D float arrays (vectors). The "ndim" attribute is used to specify the number of dimensions the arrays are allowed to have.

The corresponding python module containing the data model class is quite simple:

```
class NisSossPhotomModel(ReferenceFileModel):
    """
    A data model for NIRISS SOSS photom reference files.
    """
    schema_url = "nissoss_photom.schema"
```

2.1.10 The Structure of DataModels

Datamodels allows for the creation of a new model though the usual method of calling the __init__ method. Each type of model has its own class and schema. The schema is specified through the class variable schema_url. The schema gives the binding between the FITS header keyword and/or extension and the datamodels attribute name. The chief distinction between the two is that FITS has a flat model, datamodels supports a hierarchical data model. The typical structure of a datamodels class is that it first calls the __init__ method of the base class and then initializes the required arrays of the models with lines that look like they shouldn't do anything, for example

self.dq = self.dq

The reason why a line like the above initializes the array is that the access to the array on the right side of the assignment will initialize the array to a default value if it is not already defined and one is found in the schema. The sequence of calls is that the dot notation invokes __getattr__, which calls _make_default if the attribute is not defined, which in turn calls _make_default_array if the schema says the attribute is an array. All these methods can be found in properties.py.

The base class for Datamodels is JwstDataModel, in model_base.py. It takes several arguments, the most important of which is init, which as the name suggests, specifies how to initialize the primary data array of the model. Init is most usually the name of a file, but can be an already opened fits or asdf file, a numpy array, a shape tuple, or None. If init is a shape tuple the primary data array is initialized to its default value.

Optional arguments to __init__ can give a schema which overrides the class schema, extensions to the schema, two flags pass_invalid_values and strict_validation, which control the data validation, and numpy arrays which are used to initialized the model arrays by using parameters of the same name.

As an alternative to creating a model by initializing an object of the specific class, you can call the open function, which is in util.py. This function takes the same arguments as the __init_ method. If it is called with the name of a FITS file, it looks in the primary header for a keyword named DATAMODL that contains the name of the class to use to open the model. If that keyword is not found, checks the dimensionality of the image and uses a generic model type to open the image.

The base class for Datamodels loads the schema from the a file in the schemas subdirectory. If the base class is passed a descriptor of an already open model, it returns a shallow copy of the already open image. This is done to speed the code, as re-opening already open models is a common operation in the pipeline. If it is passed the name of a file, it peeks at the first several bytes of the file to determine the file type. This test is in filetype.py.

If the file type is a FITS file, it calls from_fits in fits_support.py to open the file. From_fits first reads the serialized version of the asdf tree stored in the asdf extension of the FITS file. It then walks through the schema, which has a tree structure and uses the fields fits_keyword and fits_hdu to locate and read items in the fits file into the asdf tree. So items in the rest of the FITS file override items in the asdf extension. It keeps track of the names of these items and then makes a pass over the FITS file and writes any keywords and hdus to another area of the asdf tree called extra_fits. Extra_fits has subtrees for each hdu. Keywords in each hdu not found in the schema are placed in the header subtree and data is placed in the data subtree. Finally, it reads the history keywords and places them in a history structure.

To write a model back to a file, call the save method on the file. It first calls validate_required to check the schema to see if all the required fields are present in the model. Then it calls the function to_fits in fits_support.py. It first creates an empty fits file and then calls a custom validator to write the contents of the asdf tree into this file. The functions called are defined by the dictionary FITS_VALIDATORS found in fits_support.py. Since the validator uses the schema as a guide, only items in the schema are added to the FITS file at this time. After saving items mentioned in the schema, to_fits then saves the contents of extra_fits, and then the history. Finally, it serializes the asdf tree and writes it to the asdf extension.

Items within a model are accessed as attribute, that is, with dot notation. The code which handles getting and setting attributes is found in properties.py. Datamodels distinguishes between items at the endpoints of the asdf tree and subtrees within the asdf tree. The former are returned as scalars or numpy arrays, depending on whether the endpoint represents a FITS keyword or data array. Subtrees are returned as nodes. A node is an object containing the subtree as well as the subschema which describes the subtree. If one tries to get an attribute that does not exist in the asdf tree, one of several things may happen. If the attribute is not mentioned in the schema, the value of the attribute is set to None. If it is in the schema and the schema has a default value, the code creates the item with the default value and then returns it. The functions that do this are _make_default and _make_default_array, which it calls. If not only the item, but the subtree containing the item is missing, the code throws an AttributeError. When an attribute representing an array is accessed, the type of the array is compared to the type in the schema and if they are different, the array is cast to the type in the schema. The same is true for numpy records, which represent rows in a FITS table. The casting is done by the function gentle_asarray in util.py.

When setting or deleting an attribute, the code validates the change. The code which does the validation can be found in validate.py. The validator checks the values of pass_invalid_values, which allows values inconsistent with the schema to be set, and strict_validation, which throws an exception if the value does not match the schema.

2.1.11 jwst.datamodels Package

Functions

open([init, guess, memmap])	Creates a DataModel from a number of different types

open

jwst.datamodels.**open**(*init=None*, *guess=True*, *memmap=False*, ***kwargs*)

Creates a DataModel from a number of different types

Parameters

init
[shape tuple, file path, file object, astropy.io.fits.HDUList,]

numpy array, dict, None

- None: A default data model with no shape
- shape tuple: Initialize with empty data of the given shape
- file path: Initialize from the given file (FITS, JSON or ASDF)
- readable file object: Initialize from the given file object
- astropy.io.fits.HDUList: Initialize from the given *~astropy.io.fits.HDUList*
- A numpy array: A new model with the data array initialized to what was passed in.
- dict: The object model tree for the data model

guess
[bool] Guess as to the model type if the model type is not specifically known from the file. If not guess and the model type is not explicit, raise a TypeError.

memmap
[bool] Turn memmap of file on or off. (default: False).

kwargs
[dict] Additional keyword arguments passed to the DataModel constructor. Some arguments are general, others are file format-specific. Arguments of note are:

- General

  validate_arrays
  [bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

- FITS

  skip_fits_update
  [bool or None] DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Returns

model
[DataModel instance]

Classes

JwstDataModel([init, schema, memmap, ...])	Parameters
ABVegaOffsetModel([init])	A data model containing offsets to convert from AB to Vega magnitudes.
AmiLgModel([init, schema, memmap, ...])	A data model for AMI LG analysis results.
AmiLgFitModel([init, schema, memmap, ...])	A data model for AMI LG analysis results.
AmiOIModel([init, schema, memmap, ...])	TODO
NRMModel([init])	A data model for Non-Redundant Mask.
FgsImgApcorrModel([init])	A data model for FGS imaging apcorr reference files.
MirImgApcorrModel([init])	A data model for MIRI imaging apcorr reference files.
NrcImgApcorrModel([init])	A data model for NIRCam imaging apcorr reference files.
NisImgApcorrModel([init])	A data model for NIRISS imaging apcorr reference files.
MirLrsApcorrModel([init])	A data model for MIRI LRS apcorr reference files.
MirMrsApcorrModel([init])	A data model for MIRI MRS apcorr reference files.
NrcWfssApcorrModel([init])	A data model for NIRCam WFSS apcorr reference files.
NisWfssApcorrModel([init])	A data model for NIRISS WFSS apcorr reference files.
NrsMosApcorrModel([init])	A data model for NIRSpec MOS apcorr reference files.
NrsFsApcorrModel([init])	A data model for NIRSpec Fixed-Slit apcorr reference files.
NrsIfuApcorrModel([init])	A data model for NIRSpec IFU apcorr reference files.
AsnModel([init])	A data model for association tables.
BarshadowModel([init])	A data model for Bar Shadow correction information.
CameraModel([init, model, input_units, ...])	A model for a reference file of type "camera".
CollimatorModel([init, model, input_units, ...])	A model for a reference file of type "collimator".
CombinedSpecModel([init, schema, memmap, ...])	A data model for combined 1D spectra.
ContrastModel([init, schema, memmap, ...])	A data model for coronagraphic contrast curve files.
CubeModel([init])	A data model for 3D image cubes.
DarkModel([init])	A data model for dark reference files.
DarkMIRIModel([init])	A data model for dark MIRI reference files.
DisperserModel([init, angle, gwa_tiltx, ...])	A model for a NIRSPEC reference file of type "disperser".
DistortionModel([init, model, input_units, ...])	A model for a reference file of type "distortion".
DistortionMRSModel([init, x_model, y_model, ...])	A model for a reference file of type "distortion" for the MIRI MRS.
DrizParsModel([init])	A data model for drizzle parameters reference tables.
EmiModel([init])	A data model to correct MIRI images for EMI contamination.
Extract1dImageModel([init, schema, memmap, ...])	A data model for the extract_1d reference image array.
Extract1dIFUModel([init])	A data model for IFU MIRI and NIRSpec extract 1d reference files.
FilteroffsetModel([init, filters, instrument])	A model for filter-dependent boresight offsets.
FlatModel([init])	A data model for 2D flat-field images.
NirspecFlatModel([init])	A data model for NIRSpec flat-field reference files.
NirspecQuadFlatModel([init])	A data model for NIRSpec flat-field files that differ by quadrant.
FOREModel([init, model, input_units, ...])	A model for a reference file of type "fore".
FPAModel([init, nrs1_model, nrs2_model])	A model for a NIRSPEC reference file of type "fpa".
FringeModel([init])	A data model for 2D fringe correction images.

continues on next page

Table 1 – continued from previous page

FringeFreqModel([init])	A data model for 2D fringe correction images.
GainModel([init])	A data model for 2D gain.
GLS_RampFitModel([init, schema, memmap, ...])	A data model for the optional output of the ramp fitting step for the GLS algorithm.
GuiderRawModel([init])	A data model for Guide Star pipeline raw data files
GuiderCalModel([init])	A data model for Guide Star pipeline calibrated files
IFUCubeModel([init])	A data model for 3D IFU cubes.
NirspecIFUCubeParsModel([init])	A data model for Nirspec ifucubepars reference files.
MiriIFUCubeParsModel([init])	A data model for MIRI mrs ifucubepars reference files.
MirMrsPtCorrModel([init])	A data model for MIRI mrs IFU across-slice corrections file.
MirMrsXArtCorrModel([init])	A data model for MIRI MRS cross-artifact corrections file.
IFUFOREModel([init, model, input_units, ...])	A model for a NIRSPEC reference file of type "ifufore".
IFUImageModel([init])	A data model for 2D IFU images.
IFUPostModel([init, slice_models])	A model for a NIRSPEC reference file of type "ifupost".
IFUSlicerModel([init, model, data])	A model for a NIRSPEC reference file of type "ifuslicer".
ImageModel([init, schema, memmap, ...])	A data model for 2D images.
IPCModel([init])	A data model for IPC kernel checking information.
IRS2Model([init, schema, memmap, ...])	A data model for the IRS2 refpix reference file.
LastFrameModel([init])	A data model for Last frame correction reference files.
Level1bModel([init, schema, memmap, ...])	A data model for raw 4D ramps level-1b products.
LinearityModel([init])	A data model for linearity correction information.
MaskModel([init])	A data model for 2D masks.
MSAModel([init, models, data])	A model for a NIRSPEC reference file of type "msa".
MultiCombinedSpecModel([init])	A data model for multi-spec images.
MultiExposureModel([init])	A data model for multi-slit images derived from numerous exposures.
MultiExtract1dImageModel([init])	A data model for extract_1d reference images.
MultiSlitModel([init])	A data model for multi-slit images.
MultiSpecModel([init])	A data model for multi-spec images.
NIRCAMGrismModel([init, displ, dispx, ...])	A model for a reference file of type "specwcs" for NIRCAM WFSS.
NIRISSGrismModel([init, displ, dispx, ...])	A model for a reference file of type "specwcs" for NIRISS grisms.
OTEModel([init, model, input_units, ...])	A model for a reference file of type "ote".
OutlierParsModel([init])	A data model for outlier detection parameters reference tables.
OutlierIFUOutputModel([init, schema, ...])	A data model for the optional output from outlier_detection_ifu step.
PathlossModel([init])	A data model for pathloss correction information.
MirLrsPathlossModel([init])	A data model for MIRI LRS pathloss correction information.
PersistenceSatModel([init])	A data model for the persistence saturation value (full well).
PixelAreaModel([init])	A data model for the pixel area map
NirspecSlitAreaModel([init])	A data model for the NIRSpec fixed-slit pixel area reference file
NirspecMosAreaModel([init])	A data model for the NIRSpec MOS pixel area reference file
NirspecIfuAreaModel([init])	A data model for the NIRSpec IFU pixel area reference file

continues on next page

Table 1 – continued from previous page

FgsImgPhotomModel([init])	A data model for FGS photom reference files.
MirImgPhotomModel([init])	A data model for MIRI imaging photom reference files.
MirLrsPhotomModel([init])	A data model for MIRI LRS photom reference files.
MirMrsPhotomModel([init])	A data model for MIRI MRS photom reference files.
NrcImgPhotomModel([init])	A data model for NIRCam imaging photom reference files.
NrcWfssPhotomModel([init])	A data model for NIRCam WFSS photom reference files.
NisImgPhotomModel([init])	A data model for NIRISS imaging photom reference files.
NisSossPhotomModel([init])	A data model for NIRISS SOSS photom reference files.
NisWfssPhotomModel([init])	A data model for NIRISS WFSS photom reference files.
NrsFsPhotomModel([init])	A data model for NIRSpec Fixed-Slit photom reference files.
NrsMosPhotomModel([init])	A data model for NIRSpec MOS and IFU photom reference files.
PsfMaskModel([init])	A data model for coronagraphic 2D PSF mask reference files
QuadModel([init])	A data model for 4D image arrays.
RampModel([init])	A data model for 4D ramps.
RampFitOutputModel([init, schema, memmap, ...])	A data model for the optional output of the ramp fitting step.
ReadnoiseModel([init])	A data model for 2D readnoise.
ReferenceFileModel([init])	A data model for reference tables
ReferenceCubeModel([init])	A data model for 3D reference images
ReferenceImageModel([init])	A data model for 2D reference images.
ReferenceQuadModel([init])	A data model for 4D reference images
RegionsModel([init, regions])	A model for a reference file of type "regions".
ResetModel([init])	A data model for reset correction reference files.
ResolutionModel([init])	A data model for Spectral Resolution parameters reference tables.
MiriResolutionModel([init])	A data model for MIRI Resolution reference files.
RSCDModel([init])	A data model for the RSCD reference file.
SaturationModel([init])	A data model for saturation checking information.
SlitDataModel([init])	A data model for 2D slit images.
SlitModel([init])	A data model for 2D images.
SpecModel([init, schema, memmap, ...])	A data model for 1D spectra.
SegmentationMapModel([init, schema, memmap, ...])	A data model for 2D segmentation maps
SossExtractModel([init, schema, memmap, ...])	A data model to hold NIRISS SOSS extraction model arrays.
SossWaveGridModel([init, schema, memmap, ...])	A data model to hold NIRISS SOSS wavelength grids.
SpecKernelModel([init])	A data model for 2D spectral kernels.
SpecProfileModel([init])	A data model for NIRISS SOSS spectral profile reference files.
SpecProfileSingleModel([init])	A data model for NIRISS SOSS spectral profile data.
SpecTraceModel([init])	A data model for NIRISS SOSS spectral trace reference files.
SpecTraceSingleModel([init])	A data model for NIRISS SOSS spectral trace data.
SpecwcsModel([init, model, input_units, ...])	A model for a reference file of type "specwcs".
StrayLightModel([init])	A data model for 2D straylight mask.
SuperBiasModel([init])	A data model for 2D super-bias images.
ThroughputModel([init])	A data model for filter throughput.

continues on next page

Table 1 – continued from previous page

TrapDensityModel([init])	A data model for the trap density of a detector, for persistence.
TrapParsModel([init])	A data model for trap capture and decay parameters.
TrapsFilledModel([init, schema, memmap, ...])	A data model for the number of traps filled for a detector, for persistence.
TsoPhotModel([init, radii])	A model for a reference file of type "tsophot".
WavelengthrangeModel([init, ...])	A model for a reference file of type "wavelengthrange".
WaveCorrModel([init, apertures])	**Parameters**
WaveMapModel([init])	A data model for NIRISS SOSS wavelength map reference files.
WaveMapSingleModel([init])	A data model for NIRISS SOSS wavelength map data.
WfssBkgModel([init])	A data model for 2D WFSS master background reference files.

JwstDataModel

class jwst.datamodels.**JwstDataModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: DataModel

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

crds_observatory	Get CRDS observatory code for this model.
schema_url	The schema URI to validate the model against.

Methods Summary

get_crds_parameters()	Get parameters used by CRDS to select references for this model.
on_init(init)	Hook invoked by the base class before returning a newly created model instance.
on_save(init)	Hook invoked by the base class before writing a model to a file (FITS or ASDF).

Attributes Documentation

crds_observatory

Get CRDS observatory code for this model.

Returns

str

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/core.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

get_crds_parameters()

Get parameters used by CRDS to select references for this model.

Returns

dict

on_init(*init*)

Hook invoked by the base class before returning a newly created model instance.

on_save(*init*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

ABVegaOffsetModel

class jwst.datamodels.**ABVegaOffsetModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model containing offsets to convert from AB to Vega magnitudes.

Parameters

abvega_offset

[*astropy.table.Table*] An astropy table containing offsets to convert from AB to Vega magnitudes. The `abvega_offset` column represents m_AB - m_Vega.

There are three types of tables, depending on the instrument, each with different column selectors. The columns names and data types are:

- FGS
  - detector: str
  - abvega_offset: float32
- NIRCam and NIRISS
  - filter: str
  - pupil: str
  - abvega_offset: float32
- MIRI
  - filter: str

- abvega_offset: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Methods Summary

validate()	Convenience function to be run when files are created.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/abvegaoffset.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

validate()
Convenience function to be run when files are created. Checks that required reference file keywords are set.

AmiLgModel

class jwst.datamodels.**AmiLgModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for AMI LG analysis results.

Parameters

fit_image
[numpy float32 array] Fitted image

resid_image
[numpy float32 array] Residual image

closure_amp_table
[numpy table] Closure amplitudes table

closure_phase_table
[numpy table] Closure phases table

fringe_amp_table
[numpy table] Fringe amplitudes table

fringe_phase_table
[numpy table] Fringe phases table

pupil_phase_table
[numpy table] Pupil phases table

solns_table
[numpy table] Solutions table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Methods Summary

get_primary_array_name()	Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/amilg.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

get_primary_array_name()
Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created. This is intended to be overridden in the subclasses if the primary array's name is not "data".

AmiLgFitModel

class jwst.datamodels.**AmiLgFitModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for AMI LG analysis results.

Parameters

centered_image
[numpy float32 array] Centered image

norm_centered_image
[numpy float32 array] Centered image normalized by data peak

fit_image
[numpy float32 array] Fitted image

norm_fit_image
[numpy float32 array] Fitted image normalized by data peak

resid_image
[numpy float32 array] Residual image

norm_resid_image
[numpy float32 array] Residual image normalized by data peak

solns_table
[numpy table] Solutions table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors

generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Methods Summary

get_primary_array_name()	Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/amilgfitmodel.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

get_primary_array_name()
Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created. This is intended to be overridden in the subclasses if the primary array's name is not "data".

AmiOlModel

class jwst.datamodels.**AmiOIModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

TODO

Parameters

TODO

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Methods Summary

get_primary_array_name()	Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created.
on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
validate()	Re-validate the model instance against its schema

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/amioi.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

get_primary_array_name()
Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created. This is intended to be overridden in the subclasses if the primary array's name is not "data".

on_save(*path=None*)
Hook invoked by the base class before writing a model to a file (FITS or ASDF).

validate()
Re-validate the model instance against its schema

NRMModel

class jwst.datamodels.**NRMModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for Non-Redundant Mask.

Parameters

nrm
[numpy float32 array] Non-Redundant Mask

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrm.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

FgsImgApcorrModel

class jwst.datamodels.**FgsImgApcorrModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for FGS imaging apcorr reference files.

Parameters

apcorr_table
[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- eefraction: float32
- radius: float32
- apcorr: float32
- skyin: float32
- skyout: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/fgsimg_apcorr.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirImgApcorrModel

class jwst.datamodels.**MirImgApcorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for MIRI imaging apcorr reference files.

Parameters

apcorr_table

[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- filter: str[12]
- subarray: str[15]
- eefraction: float32
- radius: float32
- apcorr: float32
- skyin: float32
- skyout: float32

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/mirimg_apcorr.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrcImgApcorrModel

class jwst.datamodels.**NrcImgApcorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRCam imaging apcorr reference files.

Parameters

apcorr_table

[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- filter: str[12]
- pupil: str[15]
- eefraction: float32
- radius: float32
- apcorr: float32
- skyin: float32
- skyout: float32

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrcimg_apcorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NisImgApcorrModel

class jwst.datamodels.**NisImgApcorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRISS imaging apcorr reference files.

Parameters

apcorr_table
[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- filter: str[12]
- pupil: str[15]
- eefraction: float32
- radius: float32
- apcorr: float32
- skyin: float32
- skyout: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nisimg_apcorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirLrsApcorrModel

class jwst.datamodels.**MirLrsApcorrModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for MIRI LRS apcorr reference files.

Parameters

apcorr_table
[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- subarray: str[15]
- wavelength: float32 1D array
- nelem_wl: int16
- size: uint8 1D array
- nelem_size: int16
- apcorr: float32 2D array
- apcorr_err: float32 2D array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/mirlrs_apcorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirMrsApcorrModel

class jwst.datamodels.**MirMrsApcorrModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for MIRI MRS apcorr reference files.

Parameters

apcorr_table
[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- wavelength: float32 1D array
- radius: float32 2D array
- apcorr: float32 2D array
- apcorr_err: float32 2D array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/mirmrs_apcorr.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrcWfssApcorrModel

class jwst.datamodels.**NrcWfssApcorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRCam WFSS apcorr reference files.

Parameters

apcorr_table

[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- filter: str[12]
- pupil: str[15]
- wavelength: float32 1D array
- nelem_wl: int16
- size: uint8 1D array
- nelem_size: int16
- apcorr: float32 2D array
- apcorr_err: float32 2D array

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrcwfss_apcorr.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NisWfssApcorrModel

class jwst.datamodels.**NisWfssApcorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRISS WFSS apcorr reference files.

Parameters

apcorr_table

[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- filter: str[12]
- pupil: str[15]
- wavelength: float32 1D array
- nelem_wl: int16
- size: uint8 1D array
- nelem_size: int16
- apcorr: float32 2D array
- apcorr_err: float32 2D array

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/niswfss_apcorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrsMosApcorrModel

class jwst.datamodels.**NrsMosApcorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRSpec MOS apcorr reference files.

Parameters

apcorr_table
[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- filter: str[12]
- grating: str[15]
- wavelength: float32 1D array
- nelem_wl: int16
- size: float32 2D array
- nelem_size: int16
- pixphase: float32 1D array
- apcorr: float32 3D array
- apcorr_err: float32 3D array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrsmos_apcorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrsFsApcorrModel

class jwst.datamodels.**NrsFsApcorrModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for NIRSpec Fixed-Slit apcorr reference files.

Parameters

apcorr_table
[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- filter: str[12]
- grating: str[15]
- slit: str[15]
- wavelength: float32 1D array
- nelem_wl: int16
- size: float32 2D array
- nelem_size: int16
- pixphase: float32 1D array
- apcorr: float32 3D array
- apcorr_err: float32 3D array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrsfs_apcorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrsIfuApcorrModel

class jwst.datamodels.**NrsIfuApcorrModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for NIRSpec IFU apcorr reference files.

Parameters

apcorr_table
[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- filter: str[12]
- grating: str[15]
- wavelength: float32 1D array
- radius: float32 3D array
- apcorr: float32 3D array
- apcorr_err: float32 3D array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrsifu_apcorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

AsnModel

class jwst.datamodels.**AsnModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for association tables.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.
supported_formats	

Methods Summary

parse_table()

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/asn.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

supported_formats = ['yaml', 'json', 'fits']

Methods Documentation

parse_table()

BarshadowModel

class jwst.datamodels.**BarshadowModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for Bar Shadow correction information.

Parameters

data1x1
[numpy float32 array] Bar Shadow 1x1 data array

var1x1
[numpy float32 array] Bar Shadow 1x1 correction variance

data1x3
[numpy float32 array] Bar Shadow 1x3 data array

var1x3
[numpy float32 array] Bar Shadow 1x3 correction variance

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.

- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/barshadow.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

CameraModel

class jwst.datamodels.**CameraModel**(*init=None*, *model=None*, *input_units=None*, *output_units=None*, ***kwargs*)

Bases: _SimpleModel

A model for a reference file of type "camera".

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values

[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation

[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment

[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

populate_meta()	Subclasses can overwrite this to populate specific meta keywords.

Attributes Documentation

reftype = 'camera'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/camera.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

populate_meta()
Subclasses can overwrite this to populate specific meta keywords.

CollimatorModel

class jwst.datamodels.**CollimatorModel**(*init=None*, *model=None*, *input_units=None*, *output_units=None*, ***kwargs*)

Bases: `_SimpleModel`

A model for a reference file of type "collimator".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

populate_meta()	Subclasses can overwrite this to populate specific meta keywords.

Attributes Documentation

reftype = 'collimator'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/collimator.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

populate_meta()
Subclasses can overwrite this to populate specific meta keywords.

CombinedSpecModel

class jwst.datamodels.**CombinedSpecModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for combined 1D spectra.

Parameters

spec_table
[numpy table] Combined, extracted spectral data table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/combinedspec.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

ContrastModel

class jwst.datamodels.**ContrastModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for coronagraphic contrast curve files.

Parameters

contrast_table
[numpy table] Contrast curve table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/contrast.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

CubeModel

class jwst.datamodels.**CubeModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 3D image cubes.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

zeroframe
[numpy float32 array] Zero frame array

area
[numpy float32 array] Pixel area map array

int_times
[numpy table] table of times for each integration

wavelength
[numpy float32 array] Wavelength array

var_poisson
[numpy float32 array] Integration-specific variances of slope due to Poisson noise

var_rnoise
[numpy float32 array] Integration-specific variances of slope due to read noise

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array

- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/cube.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

DarkModel

class jwst.datamodels.**DarkModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for dark reference files.

Parameters

data

[numpy float32 array] Dark current array

dq

[numpy uint32 array] 2-D data quality array for all planes

err

[numpy float32 array] Error array

dq_def

[numpy table] DQ flag definitions

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values

[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/dark.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

DarkMIRIModel

class jwst.datamodels.**DarkMIRIModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for dark MIRI reference files.

Parameters

data
[numpy float32 array] Dark current array

dq
[numpy uint32 array] 2-D data quality array for all planes

err
[numpy float32 array] Error array

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/darkMIRI.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

DisperserModel

class jwst.datamodels.**DisperserModel**(*init=None*, *angle=None*, *gwa_tiltx=None*, *gwa_tilty=None*, *kcoef=None*, *lcoef=None*, *tcoef=None*, *pref=None*, *tref=None*, *theta_x=None*, *theta_y=None*, *theta_z=None*, *groovedensity=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a NIRSPEC reference file of type “disperser”.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'disperser'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/disperser.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

DistortionModel

class jwst.datamodels.**DistortionModel**(*init=None*, *model=None*, *input_units=None*, *output_units=None*, ***kwargs*)

Bases: _SimpleModel

A model for a reference file of type "distortion".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'distortion'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/distortion.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

DistortionMRSModel

class jwst.datamodels.**DistortionMRSModel**(*init=None*, *x_model=None*, *y_model=None*, *alpha_model=None*, *beta_model=None*, *bzero=None*, *bdel=None*, *input_units=None*, *output_units=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a reference file of type "distortion" for the MIRI MRS.

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array

- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'distortion'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/distortion_mrs.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

DrizParsModel

class jwst.datamodels.**DrizParsModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for drizzle parameters reference tables.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/drizpars.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

EmiModel

class jwst.datamodels.**EmiModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model to correct MIRI images for EMI contamination.

Parameters

data

[numpy table] The reference waves to correct for MIRI EMI. A table-like object containing phase amplitude values corresponding to the appropriate frequency - Hz390: float32 1D array - Hz218a: float32 1D array - Hz218b: float32 1D array - Hz218c: float32 1D array - Hz164: float32 1D array - Hz10: float32 1D array

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'emicorr'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/emi.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

Extract1dImageModel

class jwst.datamodels.**Extract1dImageModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for the extract_1d reference image array.

Parameters

data

[numpy float32 array] 1-D extraction regions array

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/extract1dimage.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Extract1dIFUModel

class jwst.datamodels.**Extract1dIFUModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for IFU MIRI and NIRSpec extract 1d reference files.

Parameters

extract1d_params
[numpy table] Basic extract 1D parameters - region_type: ascii - subtract_background: bool - method: ascii - subpixels: int16

extract1d_table
[numpy table] extract1d parameters for varying wavelengths A table-like object containing extract 1d parameters based on wavelength - wavelength: float32 1D array - radius: float32 1D array - inner_bkg: float32 1D array - outer_bkg: float32 1D array - axis_ratio: float32 1D array - axis_pa: float32 1D array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors

generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/extract1difu.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

FilteroffsetModel

class jwst.datamodels.**FilteroffsetModel**(*init=None*, *filters=None*, *instrument=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A model for filter-dependent boresight offsets.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array

- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

populate_meta()	
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'filteroffset'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/filteroffset.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

populate_meta()

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

FlatModel

class jwst.datamodels.**FlatModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D flat-field images.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/flat.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NirspecFlatModel

class jwst.datamodels.**NirspecFlatModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRSpec flat-field reference files.

Parameters

data
[numpy float32 array] NIRSpec flat-field reference data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error estimate

wavelength
[numpy table] Table of wavelengths for image planes

flat_table
[numpy table] Table for quickly varying component of flat field

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nirspec_flat.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NirspecQuadFlatModel

class jwst.datamodels.**NirspecQuadFlatModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRSpec flat-field files that differ by quadrant.

Parameters

quadrants.items.data : numpy float32 array

quadrants.items.dq : numpy uint32 array

quadrants.items.err : numpy float32 array

quadrants.items.wavelength
[numpy table] Table of wavelengths for image planes

quadrants.items.flat_table
[numpy table] Table for quickly varying component of flat field

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nirspec_quad_flat.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

FOREModel

class jwst.datamodels.**FOREModel**(*init=None*, *model=None*, *input_units=None*, *output_units=None*, ***kwargs*)

Bases: `_SimpleModel`

A model for a reference file of type "fore".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	Subclasses can overwrite this to populate specific meta keywords.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'fore'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/fore.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)
Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()
Subclasses can overwrite this to populate specific meta keywords.

validate()
Convenience function to be run when files are created. Checks that required reference file keywords are set.

FPAModel

class jwst.datamodels.**FPAModel**(*init=None*, *nrs1_model=None*, *nrs2_model=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a NIRSPEC reference file of type "fpa".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'fpa'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/fpa.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)
Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

to_fits()
Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

`validate()`
Convenience function to be run when files are created. Checks that required reference file keywords are set.

FringeModel

`class jwst.datamodels.FringeModel(init=None, **kwargs)`

Bases: *ReferenceFileModel*

A data model for 2D fringe correction images.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/fringe.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

FringeFreqModel

class jwst.datamodels.**FringeFreqModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for 2D fringe correction images.

Parameters

rfc_freq_short : numpy table rfc_freq_medium : numpy table rfc_freq_long : numpy table max_amp : numpy table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/fringe_freq.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

GainModel

class jwst.datamodels.**GainModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D gain.

Parameters

data
[numpy float32 array] The gain

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/gain.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

GLS_RampFitModel

class jwst.datamodels.**GLS_RampFitModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for the optional output of the ramp fitting step for the GLS algorithm.

Parameters

yint
[numpy float32 array] Y-intercept

sigyint
[numpy float32 array] Sigma for Y-intercept

pedestal
[numpy float32 array] Pedestal

crmag
[numpy float32 array] CR magnitudes

sigcrmag
[numpy float32 array] Sigma for CR magnitudes

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/gls_rampfit.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

GuiderRawModel

class jwst.datamodels.**GuiderRawModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for Guide Star pipeline raw data files

Parameters

data
[numpy float32 array] The science data

err
[numpy float32 array] Error array

dq
[numpy uint32 array] Data quality array

planned_star_table
[numpy table] Planned reference star table

flight_star_table
[numpy table] Flight reference star table

pointing_table
[numpy table] Pointing table

centroid_table
[numpy table] Centroid packet table

track_sub_table
[numpy table] Track subarray data table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/guider_raw.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

GuiderCalModel

class jwst.datamodels.**GuiderCalModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for Guide Star pipeline calibrated files

Parameters

data
[numpy float32 array] The science data

err
[numpy float32 array] Error array

dq
[numpy uint32 array] Data quality array

planned_star_table
[numpy table] Planned reference star table

flight_star_table
[numpy table] Flight reference star table

pointing_table
[numpy table] Pointing table

centroid_table
[numpy table] Centroid packet table

track_sub_table
[numpy table] Track subarray data table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/guider_cal.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

IFUCubeModel

class jwst.datamodels.**IFUCubeModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 3D IFU cubes.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

weightmap
[numpy float32 array] Weight map of coverage

wavetable
[numpy table] Wavelength value for slices

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/ifucube.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NirspecIFUCubeParsModel

class jwst.datamodels.**NirspecIFUCubeParsModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for Nirspec ifucubepars reference files.

Parameters

ifucubepars_table
[numpy table] default IFU cube parameters table

ifucubepars_msm_table
[numpy table] default IFU cube msm parameters table

ifucubepars_prism_msm_wavetable
[numpy table] default IFU cube prism msm wavetable

ifucubepars_med_msm_wavetable
[numpy table] default IFU cube med resolution msm wavetable

ifucubepars_high_msm_wavetable
[numpy table] default IFU cube high resolution msm wavetable

ifucubepars_emsm_table
[numpy table] default IFU cube emsm parameters table

ifucubepars_prism_emsm_wavetable
[numpy table] default IFU cube prism emsm wavetable

ifucubepars_med_emsm_wavetable
[numpy table] default IFU cube med resolution emsm wavetable

ifucubepars_high_emsm_wavetable
[numpy table] default IFU cube high resolution emsm wavetable

ifucubepars_prism_driz_wavetable
[numpy float32 array] default IFU cube prism drizzle wavetable

ifucubepars_med_driz_wavetable
[numpy float32 array] default IFU cube med resolution drizzle wavetable

ifucubepars_high_driz_wavetable
[numpy float32 array] default IFU cube high resolution drizzle wavetable

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors

generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nirspec_ifucubepars.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MiriIFUCubeParsModel

class jwst.datamodels.**MiriIFUCubeParsModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for MIRI mrs ifucubepars reference files.

Parameters

ifucubepars_table
[numpy table] default IFU cube parameters table

ifucubepars_msm_table
[numpy table] default IFU cube msm parameters table

ifucubepars_multichannel_msm_wavetable
[numpy table] default IFU cube msm wavetable

ifucubepars_emsm_table
[numpy table] default IFU cube emsm parameters table

ifucubepars_multichannel_emsm_wavetable
[numpy table] default IFU cube emsm wavetable

ifucubepars_multichannel_driz_wavetable
[numpy float32 array] default IFU cube driz wavetable

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/miri_ifucubepars.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirMrsPtCorrModel

class jwst.datamodels.**MirMrsPtCorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for MIRI mrs IFU across-slice corrections file.

Parameters

init
[any] Any of the initializers supported by *~jwst.datamodels.DataModel*.

leakcor_table
[numpy table] IFU spectral leak correction (fractional, Jy to Jy)

tracor_table
[numpy table] IFU across slice transmission correction

wavcorr_optical_table
[numpy table] IFU across slice wavelength offset table 1

wavcorr_xslice_table
[numpy table] IFU across slice wavelength offset table 2

wavcorr_shift_table
[numpy table] IFU across slice wavelength offset table 3

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.

- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/miri_mrsptcorr.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirMrsXArtCorrModel

class jwst.datamodels.**MirMrsXArtCorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for MIRI MRS cross-artifact corrections file.

Parameters

init
[any] Any of the initializers supported by *~jwst.datamodels.DataModel*.

ch1a_table
[numpy table] Cross artifact correction parameters for Channel 1A

ch1b_table
[numpy table] Cross artifact correction parameters for Channel 1B

ch1c_table
[numpy table] Cross artifact correction parameters for Channel 1C

ch2a_table
[numpy table] Cross artifact correction parameters for Channel 2A

ch2b_table
[numpy table] Cross artifact correction parameters for Channel 2B

ch2c_table
[numpy table] Cross artifact correction parameters for Channel 2C

ch3a_table
[numpy table] Cross artifact correction parameters for Channel 3A

ch3b_table
[numpy table] Cross artifact correction parameters for Channel 3B

ch3c_table
[numpy table] Cross artifact correction parameters for Channel 3C

ch4a_table
[numpy table] Cross artifact correction parameters for Channel 4A

ch4b_table
[numpy table] Cross artifact correction parameters for Channel 4B

ch4c_table
[numpy table] Cross artifact correction parameters for Channel 4C

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/miri_mrsxartcorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

IFUFOREModel

class jwst.datamodels.**IFUFOREModel**(*init=None*, *model=None*, *input_units=None*, *output_units=None*, ***kwargs*)

Bases: _SimpleModel

A model for a NIRSPEC reference file of type "ifufore".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

populate_meta()	Subclasses can overwrite this to populate specific meta keywords.

Attributes Documentation

reftype = 'ifufore'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/ifufore.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

populate_meta()
Subclasses can overwrite this to populate specific meta keywords.

IFUImageModel

class jwst.datamodels.**IFUImageModel**(*init=None*, ***kwargs*)
Bases: *JwstDataModel*

A data model for 2D IFU images.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

zeroframe
[numpy float32 array] Zeroframe array

var_poisson
[numpy float32 array] variance due to poisson noise

var_rnoise
[numpy float32 array] variance due to read noise

wavelength
[numpy float32 array] wavelength

pathloss_point
[numpy float32 array] pathloss correction for point source

pathloss_uniform
[numpy float32 array] pathloss correction for uniform source

area
[numpy float32 array] Pixel area map array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.

- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/ifuimage.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

IFUPostModel

class jwst.datamodels.**IFUPostModel**(*init=None*, *slice_models=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A model for a NIRSPEC reference file of type "ifupost".

Parameters

init
[str] A file name.

slice_models
[dict] A dictionary with slice transforms with the following entries {"slice_N": {'linear': astropy.modeling.Model, ... 'xpoly': astropy.modeling.Model, ... 'xpoly_distortion': astropy.modeling.Model, ... 'ypoly': astropy.modeling.Model, ... 'ypoly_distortion': astropy.modeling.Model, ... }}

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'ifupost'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/ifupost.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

IFUSlicerModel

class jwst.datamodels.**IFUSlicerModel**(*init=None*, *model=None*, *data=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a NIRSPEC reference file of type "ifuslicer".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'ifuslicer'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/ifuslicer.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

ImageModel

class jwst.datamodels.**ImageModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 2D images.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

zeroframe
[numpy float32 array] Zeroframe array

var_poisson
[numpy float32 array] variance due to poisson noise

var_rnoise
[numpy float32 array] variance due to read noise

area
[numpy float32 array] Pixel area map array

pathloss_point
[numpy float32 array] Pathloss correction for point source

pathloss_uniform
[numpy float32 array] Pathloss correction for uniform source

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/image.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

IPCModel

class jwst.datamodels.**IPCModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for IPC kernel checking information.

Parameters

data
[numpy float32 array] IPC deconvolution kernel

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/ipc.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

IRS2Model

class jwst.datamodels.**IRS2Model**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for the IRS2 refpix reference file.

Parameters

irs2_table
[numpy table] Table for IRS2 refpix correction. A table with 8 columns and 2916352 (2048 * 712 * 2) rows. All values are float, but these are interpreted as alternating real and imaginary parts (real, imag, real, imag, ...) of complex values. There are four columns for ALPHA and four for BETA.

dq_table
[data quality info table] Table for identifying bad reference pixels. A table with three columns (OUTPUT, ODD_EVEN, and MASK) and eight rows.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array

- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/irs2.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

LastFrameModel

class jwst.datamodels.**LastFrameModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for Last frame correction reference files.

Parameters

data

[numpy float32 array] Last Frame Correction array

dq

[numpy uint32 array] 2-D data quality array

err

[numpy float32 array] Error array

dq_def

[numpy table] DQ flag definitions

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values

[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/lastframe.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Level1bModel

class jwst.datamodels.**Level1bModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for raw 4D ramps level-1b products.

Parameters

data
[numpy uint16 array] The science data

zeroframe
[numpy uint16 array] Zeroframe array

refout
[numpy uint16 array] Reference Output

group
[numpy table] group parameters table

int_times
[numpy table] table of times for each integration

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/level1b.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

LinearityModel

class jwst.datamodels.**LinearityModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for linearity correction information.

Parameters

coeffs
[numpy float32 array] Linearity coefficients

dq
[numpy uint32 array] Data quality flags

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Methods Summary

get_primary_array_name()	Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/linearity.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

get_primary_array_name()

Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created. This is intended to be overridden in the subclasses if the primary array's name is not "data".

MaskModel

class jwst.datamodels.**MaskModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D masks.

Parameters

dq

[numpy uint32 array] The mask

dq_def

[numpy table] DQ flag definitions

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object

- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Methods Summary

get_primary_array_name()	Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/mask.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

get_primary_array_name()

Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created. This is intended to be overridden in the subclasses if the primary array's name is not "data".

MSAModel

class jwst.datamodels.**MSAModel**(*init=None*, *models=None*, *data=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a NIRSPEC reference file of type "msa".

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'msa'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/msa.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

MultiCombinedSpecModel

class jwst.datamodels.**MultiCombinedSpecModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for multi-spec images.

This model has a special member *spec* that can be used to deal with an entire spectrum at a time. It behaves like a list:

```
>>> from stdatamodels.jwst.datamodels import CombinedSpecModel
>>> multispec_model = MultiCombinedSpecModel()
>>> multispec_model.spec.append(CombinedSpecModel())
>>> multispec_model.spec[0]
<CombinedSpecModel>
```

If *init* is a *CombinedSpecModel* instance, an empty *CombinedSpecModel* will be created and assigned to attribute *spec[0]*, and the *spec_table* attribute from the input *CombinedSpecModel* instance will be copied to the first element of *spec*. *CombinedSpecModel* objects can be appended to the *spec* attribute by using its *append* method.

Parameters

int_times
[numpy table] table of times for each integration

spec.items.spec_table
[numpy table] Extracted spectral data table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/multicombinedspec.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MultiExposureModel

class jwst.datamodels.**MultiExposureModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for multi-slit images derived from numerous exposures. The intent is that all slits in this model are of the same source, with each slit representing a separate exposure of that source.

This model has a special member *exposures* that can be used to deal with an entire slit at a time. It behaves like a list:

```
>>> from .image import ImageModel
>>> multiexposure_model = MultiExposureModel()
>>> multiexposure_model.exposures.append(ImageModel())
>>> multiexposure_model.exposures[0]
<ImageModel>
```

Also, there is an extra attribute, *meta*. This will contain the meta attribute from the exposure from which each slit has been taken.

See the module *exp_to_source* for the initial creation of these models. This is part of the Level 3 processing of multi-objection observations.

Parameters

exposures.items.data : numpy float32 array

exposures.items.dq : numpy uint32 array

exposures.items.err : numpy float32 array

exposures.items.area : numpy float32 array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

core_schema_url	
schema_url	The schema URI to validate the model against.

Attributes Documentation

core_schema_url = 'http://stsci.edu/schemas/jwst_datamodel/core.schema'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/multiexposure.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MultiExtract1dImageModel

class jwst.datamodels.**MultiExtract1dImageModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for extract_1d reference images.

This model has a special member *images* that can be used to deal with each image separately. It behaves like a list:

```
>>> from stdatamodels.jwst.datamodels import Extract1dImageModel
>>> multiextr1d_img_model = MultiExtract1dImageModel()
>>> multiextr1d_img_model.images.append(Extract1dImageModel())
>>> multiextr1d_img_model.images[0]
<Extract1dImageModelModel>
```

If *init* is a file name or an *Extract1dImageModel* instance, an empty *Extract1dImageModel* will be created and assigned to attribute *images[0]*, and the *data* attribute from the input array or *Extract1dImageModel* will be copied to the first element of *images*.

Parameters

images.items.data : numpy float32 array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/multiextract1d.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MultiSlitModel

class jwst.datamodels.**MultiSlitModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for multi-slit images.

This model has a special member *slits* that can be used to deal with an entire slit at a time. It behaves like a list:

```
>>> from stdatamodels.jwst.datamodels import SlitModel
>>> multislit_model = MultiSlitModel()
>>> multislit_model.slits.append(SlitModel())
>>> multislit_model[0]
<SlitModel>
```

If `init` is a file name or an `ImageModel` or a `SlitModel``instance, an empty ``SlitModel` will be created and assigned to attribute `slits[0]`, and the *data*, `dq`, `err`, `var_rnoise`, and `var_poisson` attributes from the input file or model will be copied to the first element of `slits`.

Parameters

slits.items.data
[numpy float32 array] The science data

slits.items.dq
[numpy uint32 array] Data quality array

slits.items.err
[numpy float32 array] Error array

slits.items.var_poisson
[numpy float32 array] variance due to poisson noise

slits.items.var_rnoise
[numpy float32 array] variance due to read noise

slits.items.wavelength
[numpy float32 array] Wavelength array, corrected for zero-point

slits.items.barshadow
[numpy float32 array] Bar shadow correction

slits.items.flatfield_point
[numpy float32 array] flatfield array for point source

slits.items.flatfield_uniform
[numpy float32 array] flatfield array for uniform source

slits.items.pathloss_point
[numpy float32 array] pathloss array for point source

slits.items.pathloss_uniform
[numpy float32 array] pathloss array for uniform source

slits.items.photom_point
[numpy float32 array] photom array for point source

slits.items.photom_uniform
[numpy float32 array] photom array for uniform source

slits.items.area
[numpy float32 array] Pixel area map array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/multislit.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MultiSpecModel

class jwst.datamodels.**MultiSpecModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for multi-spec images.

This model has a special member *spec* that can be used to deal with an entire spectrum at a time. It behaves like a list:

```
>>> from stdatamodels.jwst.datamodels import SpecModel
>>> multispec_model = MultiSpecModel()
>>> multispec_model.spec.append(SpecModel())
>>> multispec_model.spec[0]
<SpecModel>
```

If *init* is a *SpecModel* instance, an empty *SpecModel* will be created and assigned to attribute *spec[0]*, and the *spec_table* attribute from the input *SpecModel* instance will be copied to the first element of *spec*. *SpecModel* objects can be appended to the *spec* attribute by using its *append* method.

Parameters

int_times
[numpy table] table of times for each integration

spec.items.spec_table
[numpy table] Extracted spectral data table

Examples

```
>>> output_model = MultiSpecModel()
>>> spec = SpecModel()       # for the default data type
>>> for slit in input_model.slits:
...     slitname = slit.name
...     slitmodel = ExtractModel()
...     slitmodel.fromJSONFile(extref, slitname)
...     column, wavelength, countrate = slitmodel.extract(slit.data)
...     otab = np.array(zip(column, wavelength, countrate),
...                     dtype=spec.spec_table.dtype)
...     spec = datamodels.SpecModel(spec_table=otab)
...     output_model.spec.append(spec)
```

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/multispec.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NIRCAMGrismModel

class jwst.datamodels.**NIRCAMGrismModel**(*init=None*, *displ=None*, *dispx=None*, *dispy=None*, *invdispl=None*, *invdispx=None*, *invdispy=None*, *orders=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a reference file of type "specwcs" for NIRCAM WFSS.

This reference file contains the models for wave, x, and y polynomial solutions that describe dispersion through the grism.

Parameters

displ: `~astropy.modeling.Model`
Nircam Grism wavelength dispersion model

dispx
[*~astropy.modeling.Model*] Nircam Grism row dispersion model

dispy
[*~astropy.modeling.Model*] Nircam Grism column dispersion model

invdispl
[*~astropy.modeling.Model*] Nircam Grism inverse wavelength dispersion model

invdispx
[*~astropy.modeling.Model*] Nircam Grism inverse row dispersion model

invdispy
[*~astropy.modeling.Model*] Nircam Grism inverse column dispersion model

orders
[*~astropy.modeling.Model*] NIRCAM Grism orders, matched to the array locations of the dispersion models

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set.

If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'specwcs'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/specwcs_nircam_grism.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

`populate_meta()`

`to_fits()`

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

`validate()`

Convenience function to be run when files are created. Checks that required reference file keywords are set.

NIRISSGrismModel

`class jwst.datamodels.NIRISSGrismModel(`*init=None*, *displ=None*, *dispx=None*, *dispy=None*, *invdispl=None*, *orders=None*, *fwcpos_ref=None*, ***kwargs*`)`

Bases: *ReferenceFileModel*

A model for a reference file of type "specwcs" for NIRISS grisms.

Parameters

displ: `~astropy.modeling.Model`
NIRISS Grism wavelength dispersion model

dispx
[*~astropy.modeling.Model*] NIRISS Grism row dispersion model

dispy
[*~astropy.modeling.Model*] NIRISS Grism column dispersion model

invdispl
[*~astropy.modeling.Model*] NIRISS Grism inverse wavelength dispersion model

invdispx
[*~astropy.modeling.Model*] NIRISS Grism inverse row dispersion model

invdispy
[*~astropy.modeling.Model*] NIRISS Grism inverse column dispersion model

orders
[*~astropy.modeling.Model*] NIRISS Grism orders, matched to the array locations of the dispersion models

fwcpos_ref
[float] The reference value for the filter wheel position

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'specwcs'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/specwcs_niriss_grism.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

populate_meta()

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

OTEModel

class jwst.datamodels.**OTEModel**(*init=None*, *model=None*, *input_units=None*, *output_units=None*, ***kwargs*)

Bases: _SimpleModel

A model for a reference file of type "ote".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

populate_meta()	Subclasses can overwrite this to populate specific meta keywords.

Attributes Documentation

reftype = 'ote'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/ote.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

populate_meta()
Subclasses can overwrite this to populate specific meta keywords.

OutlierParsModel

class jwst.datamodels.**OutlierParsModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for outlier detection parameters reference tables.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/outlierpars.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

OutlierIFUOutputModel

class jwst.datamodels.**OutlierIFUOutputModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for the optional output from outlier_detection_ifu step.

In the parameter definitions below, *n* is the number of exposures, *ny* and *nx* are the height and width of the image.

Parameters

diffarr
[numpy float32 array (n, ny, nx)] Minimum difference array for all the data

minarr
[numpy float32 array (ny, nx)] Final combined minimum difference array

normarr
[numpy float32 array (ny, nx)] Normalized minarr

minnorm
[numpy float32 array (ny, nx)] minarr/normarr

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/outlierifuoutput.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

PathlossModel

class jwst.datamodels.**PathlossModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for pathloss correction information.

Parameters

apertures.items.pointsource_data
[numpy float32 array] Point source pathloss

apertures.items.pointsource_err
[numpy float32 array] Point source pathloss variance

apertures.items.uniform_data
[numpy float32 array] Uniform source pathloss

apertures.items.uniform_err
[numpy float32 array] Uniform source pathloss variance

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/pathloss.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirLrsPathlossModel

class jwst.datamodels.**MirLrsPathlossModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for MIRI LRS pathloss correction information.

Parameters

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/mirlrs_pathloss.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

PersistenceSatModel

class jwst.datamodels.**PersistenceSatModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for the persistence saturation value (full well).

Parameters

data
[numpy float32 array] Persistence saturation threshold

dq
[numpy uint32 array] data quality array

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/persat.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

PixelAreaModel

class jwst.datamodels.**PixelAreaModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for the pixel area map

Parameters

data

[numpy float32 array] The pixel area array

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values

[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation

[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/pixelarea.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NirspecSlitAreaModel

class jwst.datamodels.**NirspecSlitAreaModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for the NIRSpec fixed-slit pixel area reference file

Parameters

area_table
[numpy table] NIRSpec fixed-slit pixel area table A table-like object containing row selection criteria made up of the slit id and the pixel area values associated with the slits.

- slit_id: str[15]
- pixarea: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nirspec_area_slit.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NirspecMosAreaModel

class jwst.datamodels.**NirspecMosAreaModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for the NIRSpec MOS pixel area reference file

Parameters

area_table
[numpy table] NIRSpec MOS pixel area table A table-like object containing row selection criteria made up of MOS shutter parameters and the pixel area values associated with the shutters.

- quadrant: int16
- shutter_x: int16
- shutter_y: int16
- pixarea: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object

- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nirspec_area_mos.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NirspecIfuAreaModel

class jwst.datamodels.**NirspecIfuAreaModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for the NIRSpec IFU pixel area reference file

Parameters

area_table

[numpy table] NIRSpec IFU pixel area table A table-like object containing row selection criteria made up of IFU slice id and the pixel area values associated with the slices.

- slice_id: int16
- pixarea: float32

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nirspec_area_ifu.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

FgsImgPhotomModel

class jwst.datamodels.**FgsImgPhotomModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for FGS photom reference files.

Parameters

phot_table
[numpy table] Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- photmjsr: float32
- uncertainty: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors

generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/fgsimg_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirImgPhotomModel

class jwst.datamodels.**MirImgPhotomModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for MIRI imaging photom reference files.

Parameters

phot_table
[numpy table]

Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- subarray: str[15]
- photmjsr: float32

- uncertainty: float32

timecoeff
[numpy table]

Table with the coefficients for the time-dependent correction.

- amplitude: float32
- tau: float32
- t0: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/mirimg_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirLrsPhotomModel

class jwst.datamodels.**MirLrsPhotomModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for MIRI LRS photom reference files.

Parameters

phot_table
[numpy table]

Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- subarray: str[15]
- photmjsr: float32
- uncertainty: float32
- nelem: int16
- wavelength: float32[*]
- relresponse: float32[*]

- reluncertainty: float32[*]

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/mirlrs_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirMrsPhotomModel

class jwst.datamodels.**MirMrsPhotomModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for MIRI MRS photom reference files.

Parameters

init
[any] Any of the initializers supported by *~jwst.datamodels.DataModel*.

data
[numpy array] An array-like object containing the pixel-by-pixel conversion values in units of (MJy / pixel) / (DN / sec).

err
[numpy array] An array-like object containing the uncertainties in the conversion values, in the same units as the data array.

dq
[numpy array] An array-like object containing bit-encoded data quality flags, indicating problem conditions for values in the data array.

dq_def
[numpy array] A table-like object containing the data quality definitions table.

pixsiz
[numpy array] An array-like object containing pixel-by-pixel size values, in units of square arcseconds (arcsec^2).

timecoeff_ch1
[numpy table] A table of time and wavelength dependent throughput corrections for channel 1

timecoeff_ch2
[numpy table] A table of time and wavelength dependent throughput corrections for channel 2

timecoeff_ch3
[numpy table] A table of time and wavelength dependent throughput corrections for channel 3

timecoeff_ch4
[numpy table] A table of time and wavelength dependent throughput corrections for channel 4

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/mirmrs_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrcImgPhotomModel

class jwst.datamodels.**NrcImgPhotomModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for NIRCam imaging photom reference files.

Parameters

phot_table
[numpy table] Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- pupil: str[12]
- photmjsr: float32
- uncertainty: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)

- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrcimg_photom.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrcWfssPhotomModel

class jwst.datamodels.**NrcWfssPhotomModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRCam WFSS photom reference files.

Parameters

phot_table

[numpy table] Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- pupil: str[15]
- order: int16
- photmjsr: float32
- uncertainty: float32
- nelem: int16
- wavelength: float32[*]
- relresponse: float32[*]
- reluncertainty: float32[*]

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.

- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrcwfss_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NisImgPhotomModel

class jwst.datamodels.**NisImgPhotomModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for NIRISS imaging photom reference files.

Parameters

phot_table
[numpy table] Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- pupil: str[12]
- photmjsr: float32
- uncertainty: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nisimg_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NisSossPhotomModel

class jwst.datamodels.**NisSossPhotomModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for NIRISS SOSS photom reference files.

Parameters

phot_table
[numpy table] Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- pupil: str[15]
- order: int16
- photmj: float32
- uncertainty: float32
- nelem: int16
- wavelength: float32[*]
- relresponse: float32[*]
- reluncertainty: float32[*]

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nissoss_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NisWfssPhotomModel

class jwst.datamodels.**NisWfssPhotomModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for NIRISS WFSS photom reference files.

Parameters

phot_table
[numpy table] Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- pupil: str[15]
- order: int16
- photmjsr: float32
- uncertainty: float32
- nelem: int16
- wavelength: float32[*]
- relresponse: float32[*]
- reluncertainty: float32[*]

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/niswfss_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrsFsPhotomModel

class jwst.datamodels.**NrsFsPhotomModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRSpec Fixed-Slit photom reference files.

Parameters

phot_table
[numpy table] Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- grating: str[15]
- slit: str[15]
- photmj: float32
- uncertainty: float32
- nelem: int16
- wavelength: float32[*]
- relresponse: float32[*]
- reluncertainty: float32[*]

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrsfs_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrsMosPhotomModel

class jwst.datamodels.**NrsMosPhotomModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for NIRSpec MOS and IFU photom reference files.

Parameters

phot_table
[numpy table] Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- grating: str[15]
- photmj: float32
- uncertainty: float32
- nelem: int16
- wavelength: float32[*]
- relresponse: float32[*]
- reluncertainty: float32[*]

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set.

If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrsmos_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

PsfMaskModel

class jwst.datamodels.**PsfMaskModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for coronagraphic 2D PSF mask reference files

Parameters

data
[numpy float32 array] The PSF mask

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/psfmask.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

QuadModel

class jwst.datamodels.**QuadModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 4D image arrays.

Parameters

data

[numpy float32 array] The science data

dq

[numpy uint32 array] Data quality array

err

[numpy float32 array] Error array

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/quad.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

RampModel

class jwst.datamodels.**RampModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 4D ramps.

Parameters

data
[numpy float32 array] The science data

pixeldq
[numpy uint32 array] 2-D data quality array for all planes

groupdq
[numpy uint8 array] 4-D data quality array for each plane

err
[numpy float32 array] Error array

zeroframe
[numpy float32 array] Zeroframe array

group
[numpy table] group parameters table

int_times
[numpy table] table of times for each integration

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/ramp.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

RampFitOutputModel

class jwst.datamodels.**RampFitOutputModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for the optional output of the ramp fitting step.

In the parameter definitions below, *n_int* is the number of integrations, *max_seg* is the maximum number of segments that were fit, *nreads* is the number of reads in an integration, and *ny* and *nx* are the height and width of the image.

Parameters

slope
[numpy float32 array (n_int, max_seg, ny, nx)] Segment-specific slope

sigslope
[numpy float32 array (n_int, max_seg, ny, nx)] Sigma for segment-specific slope

var_poisson
[numpy float32 array (n_int, max_seg, ny, nx)] Variance due to poisson noise for segment-specific slope

var_rnoise
[numpy float32 array (n_int, max_seg, ny, nx)] Variance due to read noise for segment-specific slope

yint
[numpy float32 array (n_int, max_seg, ny, nx)] Segment-specific y-intercept

sigyint
[numpy float32 array (n_int, max_seg, ny, nx)] Sigma for segment-specific y-intercept

pedestal
[numpy float32 array (n_int, max_seg, ny, nx)] Pedestal array

weights
[numpy float32 array (n_int, max_seg, ny, nx)] Weights for segment-specific fits

crmag
[numpy float32 array (n_int, max_seg, ny, nx)] Approximate CR magnitudes

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array

- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/rampfitoutput.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

ReadnoiseModel

class jwst.datamodels.**ReadnoiseModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D readnoise.

Parameters

data

[numpy float32 array] Read noise

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values

[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation

[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/readnoise.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

ReferenceFileModel

class jwst.datamodels.**ReferenceFileModel**(*init=None*, ***kwargs*)
Bases: *JwstDataModel*

A data model for reference tables

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object

- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Methods Summary

print_err(message)	
save(path[, dir_path])	Save data model.
validate()	Convenience function to be run when files are created.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/referencefile.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

print_err(*message*)

save(*path*, *dir_path=None*, **args*, ***kwargs*)

Save data model. If the 'dq' and 'dq_def' exist they need special handling.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

ReferenceCubeModel

class jwst.datamodels.**ReferenceCubeModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 3D reference images

Parameters

data

[numpy float32 array] The science data

dq

[numpy uint32 array] Data quality array

err

[numpy float32 array] Error array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/referencecube.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

ReferenceImageModel

class jwst.datamodels.**ReferenceImageModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D reference images.

Reference image data model.

Parameters

data

[numpy float32 array] The science data

dq

[numpy uint32 array] Data quality array

err

[numpy float32 array] Error array

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/referenceimage.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

ReferenceQuadModel

class jwst.datamodels.**ReferenceQuadModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 4D reference images

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors

generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/referencequad.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

RegionsModel

class jwst.datamodels.**RegionsModel**(*init=None*, *regions=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a reference file of type "regions".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array

- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'regions'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/regions.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

ResetModel

class jwst.datamodels.**ResetModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for reset correction reference files.

Parameters

data
[numpy float32 array] Reset Correction array

dq
[numpy uint32 array] 2-D data quality array for each integration

err
[numpy float32 array] Error array

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/reset.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

ResolutionModel

class jwst.datamodels.**ResolutionModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for Spectral Resolution parameters reference tables.

Parameters

data
[numpy float32 array] Resolving Power table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array

- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/resolution.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MiriResolutionModel

class jwst.datamodels.**MiriResolutionModel**(*init=None*, ***kwargs*)

Bases: *ResolutionModel*

A data model for MIRI Resolution reference files.

Parameters

data

[numpy table] Resolving Power table A table containing resolving power of the MRS. THe table consist of 11 columns and 12 rows. Each row corresponds to a band. The columns give the name of band, central wavelength, and polynomial coefficients (a,b,c) needed to obtain the limits and average value of the spectral resolution.

psf_fwhm_alpha_table

[table] PSF FWHM Alpha A table with 5 columns. Column 1 gives the cutoff wavelength where the polynomials describing alpha FWHM change. Columns 2 and 3 give the polynomial coefficients (a,b) describing alpha FWHM for wavelengths shorter than cutoff. Columns 4 and 5 give the polynomial coefficients (a,b) describing alpha FWHM for wavelengths longer than the cutoff.

psf_fwhm_beta_table

[table] PSF FWHM Beta A table with 5 columns. Column 1 gives the cutoff wavelength where the polynomials describing alpha FWHM change. Columns 2 and 3 give the polynomial coefficients (a,b) describing beta FWHM for wavelengths shorter than cutoff. Columns 4 and 5 give the polynomial coefficients (a,b) describing beta FWHM for wavelengths longer than the cutoff.

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/miri_resolution.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

RSCDModel

class jwst.datamodels.**RSCDModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for the RSCD reference file.

Parameters

rscd_group_skip_table
[numpy table] Reference table for RSCD correction baseline correction A table with 3 columns that set the number of groups to skip for each subarray and readpatt

rscd_gen_table
[numpy table] Reference table for RSCD correction enhanced correction A table with 5 columns that sets up general parameters for the enhanced correction

rscd_int1_table
[numpy table] Reference table for RSCD correction enhanced correction A table with 7 columns that sets up correction parameters for the enhanced correction for integration 1 based on subarray, readpatt, even or odd row

rscd_int2_table
[numpy table] Reference table for RSCD correction enhanced correction A table with 7 columns that sets up correction parameters for the enhanced correction for integration 2 based on subarray, readpatt, even or odd row

rscd_int3_table
[numpy table] Reference table for RSCD correction enhanced correction A table with 7 columns that sets up correction parameters for the enhanced correction for integration 3 based on subarray, readpatt, even or odd row

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/rscd.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SaturationModel

class jwst.datamodels.**SaturationModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for saturation checking information.

Parameters

data
[numpy float32 array] Saturation threshold

dq
[numpy uint32 array] 2-D data quality array for all planes

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors

generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/saturation.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SlitDataModel

class jwst.datamodels.**SlitDataModel**(*init=None*, ***kwargs*)
Bases: *JwstDataModel*

A data model for 2D slit images.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

var_poisson
[numpy float32 array] variance due to poisson noise

var_rnoise
[numpy float32 array] variance due to read noise

var_flat
[numpy float32 array] variance due to flat

wavelength
[numpy float32 array] Wavelength array, corrected for zero-point

barshadow
[numpy float32 array] Bar shadow correction

flatfield_point
[numpy float32 array] flatfield array for point source

flatfield_uniform
[numpy float32 array] flatfield array for uniform source

pathloss_point
[numpy float32 array] pathloss array for point source

pathloss_uniform
[numpy float32 array] pathloss array for uniform source

photom_point
[numpy float32 array] photom array for point source

photom_uniform
[numpy float32 array] photom array for uniform source

area
[numpy float32 array] Pixel area map array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the meta-

data. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/slitdata.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SlitModel

class jwst.datamodels.**SlitModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 2D images.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

var_poisson
[numpy float32 array] variance due to poisson noise

var_rnoise
[numpy float32 array] variance due to read noise

var_flat
[numpy float32 array] variance due to flat

wavelength
[numpy float32 array] Wavelength array, corrected for zero-point

barshadow
[numpy float32 array] Bar shadow correction

flatfield_point
[numpy float32 array] flatfield array for point source

flatfield_uniform
[numpy float32 array] flatfield array for uniform source

pathloss_point
[numpy float32 array] pathloss array for point source

pathloss_uniform
[numpy float32 array] pathloss array for uniform source

photom_point
[numpy float32 array] photom array for point source

photom_uniform
[numpy float32 array] photom array for uniform source

area
[numpy float32 array] Pixel area map array

int_times
[numpy table] table of times for each integration

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/slit.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SpecModel

class jwst.datamodels.**SpecModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 1D spectra.

Parameters

spec_table

[numpy table] Extracted spectral data table A table with at least four columns: wavelength, flux, an error estimate for the flux, and data quality flags.

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/spec.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SegmentationMapModel

class jwst.datamodels.**SegmentationMapModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 2D segmentation maps

Parameters

data
[numpy uint32 array] The segmentation map

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/segmap.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SossExtractModel

class jwst.datamodels.**SossExtractModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model to hold NIRISS SOSS extraction model arrays. For each order, stores the model trace per integration and aperture pixel weights for each order extraction.

This model is written to explicitly handle each of the three orders.

Parameters

order1
[numpy float32 array] 3-D array of the 2-D model trace for each integration, for spectral order 1

order2
[numpy float32 array] 3-D array of the 2-D model trace for each integration, for spectral order 2

order3
[numpy float32 array] 3-D array of the 2-D model trace for each integration, for spectral order 3

aperture1
[numpy float32 array] 2-D array storing the pixel weights for box-extracting spectral order 1

aperture2
[numpy float32 array] 2-D array storing the pixel weights for box-extracting spectral order 2

aperture3
[numpy float32 array] 2-D array storing the pixel weights for box-extracting spectral order 3

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/sossextractmodel.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SossWaveGridModel

class jwst.datamodels.**SossWaveGridModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model to hold NIRISS SOSS wavelength grids. This 1-D array of wavelengths can be saved from a processing run and applied to future input products.

Parameters

wavegrid
[numpy float32 array] 1-D array of the wavelengths corresponding to the ATOCA fit.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array

- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/sosswavegrid.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SpecKernelModel

class jwst.datamodels.**SpecKernelModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D spectral kernels.

Parameters

wavelengths
[numpy float32 array] Wavelengths

kernels
[numpy float32 array] Kernel values

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/speckernel.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SpecProfileModel

class jwst.datamodels.**SpecProfileModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRISS SOSS spectral profile reference files.

This model has a special member *profile* that can be used to deal with an entire spectral profile at a time. It behaves like a list:

```
>>> from stdatamodels.jwst.datamodels import SpecProfileSingleModel
>>> specprofile_model = SpecProfileModel()
>>> specprofile_model.profile.append(SpecProfileSingleModel())
>>> specprofile_model.profile[0]
<SpecProfileSingleModel>
```

If *init* is a *SpecProfileSingleModel* instance, an empty *SpecProfileSingleModel* will be created and assigned to attribute *profile[0]*, and the *data* attribute from the input *SpecProfileSingleModel* instance will be copied to the first element of *profile*. *SpecProfileSingleModel* objects can be appended to the *profile* attribute by using its *append* method.

Parameters

profile.items.data
[numpy array] Spectral profile data

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/specprofile.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SpecProfileSingleModel

class jwst.datamodels.**SpecProfileSingleModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for NIRISS SOSS spectral profile data.

Parameters

data
[numpy float32 array] Spectral profile values

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array

- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/specprofilesingle.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SpecTraceModel

class jwst.datamodels.**SpecTraceModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRISS SOSS spectral trace reference files.

This model has a special member *trace* that can be used to deal with an entire spectral trace at a time. It behaves like a list:

```
>>> from stdatamodels.jwst.datamodels import SpecTraceSingleModel
>>> spectrace_model = SpecTraceModel()
>>> spectrace_model.trace.append(SpecTraceSingleModel())
>>> spectrace_model.trace[0]
<SpecTraceSingleModel>
```

If *init* is a *SpecTraceSingleModel* instance, an empty *SpecTraceSingleModel* will be created and assigned to attribute *trace[0]*, and the *data* attribute from the input *SpecTraceSingleModel* instance will be copied to the first element of *trace*. *SpecTraceSingleModel* objects can be appended to the *trace* attribute by using its *append* method.

Parameters

trace.items.data

[numpy table] Spectral trace data

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/spectrace.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SpecTraceSingleModel

class jwst.datamodels.**SpecTraceSingleModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRISS SOSS spectral trace data.

Parameters

data
[numpy table] Trace values

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/spectracesingle.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SpecwcsModel

class jwst.datamodels.**SpecwcsModel**(*init=None*, *model=None*, *input_units=None*, *output_units=None*, ***kwargs*)

Bases: _SimpleModel

A model for a reference file of type "specwcs".

Notes

For NIRISS and NIRCAM WFSS modes the specwcs file is used during extract_2D. See NIRCAMGrismModel and NIRISSGrismModel.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array

- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'specwcs'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/specwcs.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

StrayLightModel

class jwst.datamodels.**StrayLightModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D straylight mask.

Parameters

data

[numpy uint8 array] Straylight mask

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/straylight.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SuperBiasModel

class jwst.datamodels.**SuperBiasModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D super-bias images.

Parameters

data

[numpy float32 array] The science data

dq

[numpy uint32 array] Data quality array

err

[numpy float32 array] Error array

dq_def

[numpy table] DQ flag definitions

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values

[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/superbias.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

ThroughputModel

class jwst.datamodels.**ThroughputModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for filter throughput.

Parameters

filter_table
[numpy table] Filter throughput table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/throughput.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

TrapDensityModel

class jwst.datamodels.**TrapDensityModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for the trap density of a detector, for persistence.

Parameters

data
[numpy float32 array] Trap density

dq
[numpy uint32 array] data quality array

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/trapdensity.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

TrapParsModel

class jwst.datamodels.**TrapParsModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for trap capture and decay parameters.

Parameters

trappars_table

[numpy table] Trap capture and decay parameters A table with three columns for trap-capture parameters and one column for the trap-decay parameter. Each row of the table is for a different trap family.

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values

[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation

[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/trappars.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

TrapsFilledModel

class jwst.datamodels.**TrapsFilledModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for the number of traps filled for a detector, for persistence.

Parameters

data
[numpy float32 array] Traps filled The map of the number of traps filled over the detector, with one plane for each "trap family."

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/trapsfilled.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

TsoPhotModel

class jwst.datamodels.**TsoPhotModel**(*init=None*, *radii=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A model for a reference file of type "tsophot".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'tsophot'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/tsophot.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

WavelengthrangeModel

class jwst.datamodels.**WavelengthrangeModel**(*init=None*, *wrange_selector=None*, *wrange=None*, *order=None*, *extract_orders=None*, *wunits=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a reference file of type "wavelengthrange".

The model is used by MIRI, NIRSPEC, NIRCAM, and NIRISS.

Parameters

wrange
[list] Contains a list of [order, filter, min wave, max wave]

order
[list] A list of orders that are available and described in the file

extract_orders
[list] A list of filters and the orders that should be extracted by default

wunits
[~*astropy.units*] The units for the wavelength data

Parameters

init
[str, tuple, ~*astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

get_wfss_wavelength_range(filter, orders)	Retrieve the wavelength range for a WFSS observation.
on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'wavelengthrange'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/wavelengthrange.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

get_wfss_wavelength_range(*filter*, *orders*)

Retrieve the wavelength range for a WFSS observation.

Parameters

filter
[str] Filter for which to retrieve the wavelength range.

orders
[list] List of spectral orders

Returns

wave_range
[dict] Pairs of {order: (wave_min, wave_max)} for each order and the specific filter.

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

`validate()`

Convenience function to be run when files are created. Checks that required reference file keywords are set.

WaveCorrModel

`class jwst.datamodels.WaveCorrModel(init=None, apertures=None, **kwargs)`

Bases: *ReferenceFileModel*

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

aperture_names	
reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	
validate()	Convenience function to be run when files are created.

Attributes Documentation

aperture_names

reftype = 'wavecorr'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/wavecorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

WaveMapModel

class jwst.datamodels.**WaveMapModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRISS SOSS wavelength map reference files.

This model has a special member *map* that can be used to deal with an entire wavelength map at a time. It behaves like a list:

```
>>> from stdatamodels.jwst.datamodels import WaveMapSingleModel, WaveMapModel
>>> wavemap_model = WaveMapModel()
>>> wavemap_model.map.append(WaveMapSingleModel())
>>> wavemap_model.map[0]
<WaveMapSingleModel>
```

If *init* is a *WaveMapSingleModel* instance, an empty *WaveMapSingleModel* will be created and assigned to attribute *map[0]*, and the *data* attribute from the input *WaveMapSingleModel* instance will be copied to the first element of *map*. *WaveMapSingleModel* objects can be appended to the *map* attribute by using its *append* method.

Parameters

map.items.data
[numpy data array] Wavelength map data

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/wavemap.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

WaveMapSingleModel

class jwst.datamodels.**WaveMapSingleModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRISS SOSS wavelength map data.

Parameters

data

[numpy float32 array] Wavelength values

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values

[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation

[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/wavemapsingle.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

WfssBkgModel

class jwst.datamodels.**WfssBkgModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D WFSS master background reference files.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

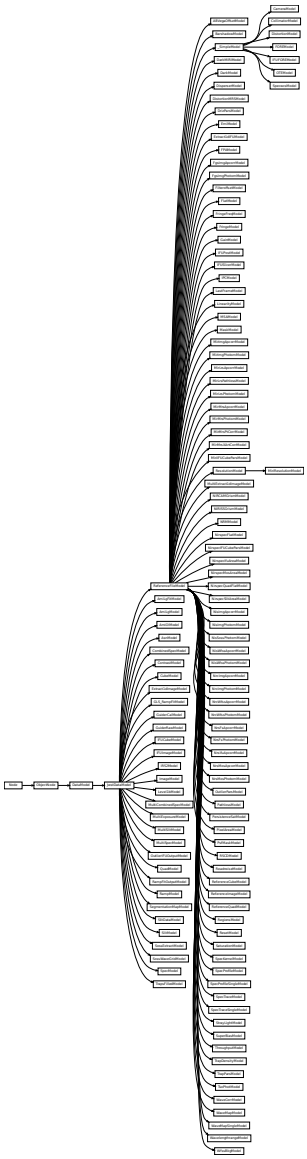
schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/wfssbkg.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Class Inheritance Diagram



2.2 Transforms

2.2.1 ASDF Schema Definitions

This package defines ASDF schemas that are used for validating the representation of transforms in the ASDF format. These schemas contain useful documentation about the associated types, and can also be used by other implementations that wish to interoperate with these transform definitions.

CHAPTER

THREE

API

3.1 stdatamodels API

3.1.1 stdatamodels Package

Classes

DataModel([init, schema, memmap, ...])	Base class of all of the data models.

DataModel

class stdatamodels.**DataModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: `ObjectNode`

Base class of all of the data models.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

crds_observatory	Get the CRDS observatory code for this model.
history	Get the history as a list of entries
override_handle	override_handle identifies in-memory models where a filepath would normally be used.
schema	
schema_url	The schema URI to validate the model against.
shape	

Methods Summary

add_schema_entry(position, new_schema)	Extend the model's schema by placing the given new_schema at the given dot-separated position in the tree.
clone(target, source[, deepcopy, memo])	
close()	
copy([memo])	Returns a deep copy of this model.
extend_schema(new_schema)	Extend the model's schema using the given schema, by combining it in an "allOf" array.
find_fits_keyword(keyword[, return_result])	Utility function to find a reference to a FITS keyword in this model's schema.
from_asdf(init[, schema])	Load a data model from an ASDF file.
from_fits(init[, schema])	Load a model from a FITS file.
get_crds_parameters()	Get the parameters used by CRDS to select references for this model.
get_fileext()	
get_fits_wcs([hdu_name, hdu_ver, key])	Get a *astropy.wcs.WCS* object created from the FITS WCS information in the model.
get_primary_array_name()	Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created.
getarray_noinit(attribute)	Retrieve array but without initilization
info(*args, **kwargs)	Print a rendering of this file's tree to stdout.
items()	Iterates over all of the schema items in a flat way.
keys()	Iterates over all of the schema keys in a flat way.
on_init(init)	Hook used to customize model attributes at the end of `__init__`.
on_save([path])	This is a hook that is called just before saving the file.
open_asdf([init, ignore_version_mismatch, ...])	Open an asdf object from a filename or create a new asdf object
read([init, schema, memmap, ...])	**Parameters**
save(path[, dir_path])	Save to either a FITS or ASDF file, depending on the path.
search(*args, **kwargs)	Search this file's tree.
search_schema(substring)	Utility function to search the metadata schema for a particular phrase.
set_fits_wcs(wcs[, hdu_name])	Sets the FITS WCS information on the model using the given *astropy.wcs.WCS* object.
to_asdf(init, *args, **kwargs)	Write a data model to an ASDF file.
to_fits(init, *args, **kwargs)	Write a data model to a FITS file.
to_flat_dict([include_arrays])	Returns a dictionary of all of the schema items as a flat dictionary.
update(d[, only, extra_fits])	Updates this model with the metadata elements from another model.
validate()	Re-validate the model instance against its schema
values()	Iterates over all of the schema values in a flat way.

continues on next page

Table 1 – continued from previous page

write(path, *args, **kwargs)

Attributes Documentation

crds_observatory

Get the CRDS observatory code for this model.

Returns

str

history

Get the history as a list of entries

override_handle

override_handle identifies in-memory models where a filepath would normally be used.

schema

schema_url = None

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

shape

Methods Documentation

add_schema_entry(*position*, *new_schema*)

Extend the model's schema by placing the given new_schema at the given dot-separated position in the tree.

Parameters

position
[str] Dot separated string indicating the position, e.g. `meta.instrument.name`.

new_schema
[dict] Schema tree.

static clone(*target*, *source*, *deepcopy=False*, *memo=None*)

close()

copy(*memo=None*)

Returns a deep copy of this model.

extend_schema(*new_schema*)

Extend the model's schema using the given schema, by combining it in an "allOf" array.

Parameters

new_schema
[dict] Schema tree.

`find_fits_keyword`(*keyword*, *return_result=True*)

Utility function to find a reference to a FITS keyword in this model's schema. This is intended for interactive use, and not for use within library code.

Parameters

keyword
[str] A FITS keyword name.

Returns

locations
[list of str] If *return_result* is *True*, a list of the locations in the schema where this FITS keyword is used. Each element is a dot-separated path.

`classmethod from_asdf`(*init*, *schema=None*, ***kwargs*)

Load a data model from an ASDF file.

Parameters

init
[str, file object, *~asdf.AsdfFile*]

- str : file path: initialize from the given file
- readable file object: Initialize from the given file object
- *~asdf.AsdfFile* : Initialize from the given`~asdf.AsdfFile`.

schema
Same as for *__init__*

kwargs
[dict] Aadditional arguments passed to lower level functions

Returns

model
[*~jwst.datamodels.DataModel* instance] A data model.

`classmethod from_fits`(*init*, *schema=None*, ***kwargs*)

Load a model from a FITS file.

Parameters

init
[file path, file object, astropy.io.fits.HDUList]

- file path: Initialize from the given file
- readable file object: Initialize from the given file object
- astropy.io.fits.HDUList: Initialize from the given *~astropy.io.fits.HDUList*.

schema
[dict, str] Same as for *__init__*

kwargs
[dict] Aadditional arguments passed to lower level functions.

Returns

model
[*~jwst.datamodels.DataModel*] A data model.

`get_crds_parameters()`

Get the parameters used by CRDS to select references for this model.

Returns

dict

`get_fileext()`

`get_fits_wcs(hdu_name='SCI', hdu_ver=1, key=' ')`

Get a *astropy.wcs.WCS* object created from the FITS WCS information in the model.

Note that modifying the returned WCS object will not modify the data in this model. To update the model, use *set_fits_wcs*.

Parameters

hdu_name
[str, optional] The name of the HDU to get the WCS from. This must use named HDU's, not numerical order HDUs. To get the primary HDU, pass `'PRIMARY'`.

key
[str, optional] The name of a particular WCS transform to use. This may be either `' '` or `'A'-'Z'` and corresponds to the `"a"` part of the `CTYPEia` cards. *key* may only be provided if *header* is also provided.

hdu_ver: int, optional
The extension version. Used when there is more than one extension with the same name. The default value, 1, is the first.

Returns

wcs
[*astropy.wcs.WCS* or *pywcs.WCS* object] The type will depend on what libraries are installed on this system.

`get_primary_array_name()`

Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created. This is intended to be overridden in the subclasses if the primary array's name is not "data".

`getarray_noinit(attribute)`

Retrieve array but without initilization

Arrays initialize when directly referenced if they had not previously been initialized. This circumvents the initialization and instead raises *AttributeError*.

Parameters

attribute
[str] The attribute to retrieve.

Returns

value
[object] The value of the attribute.

Raises

AttributeError
If the attribute does not exist.

info(**args*, ***kwargs*)

Print a rendering of this file's tree to stdout.

Parameters

max_rows
[int, tuple, or None, optional] Maximum number of lines to print. Nodes that cannot be displayed will be elided with a message. If int, constrain total number of displayed lines. If tuple, constrain lines per node at the depth corresponding to the tuple index. If None, display all lines.

max_cols
[int or None, optional] Maximum length of line to print. Nodes that cannot be fully displayed will be truncated with a message. If int, constrain length of displayed lines. If None, line length is unconstrained.

show_values
[bool, optional] Set to False to disable display of primitive values in the rendered tree.

items()

Iterates over all of the schema items in a flat way.

Each element is a pair (*key*, *value*). Each *key* is a dot-separated name. For example, the schema element *meta.observation.date* will end up in the result as:

```
("meta.observation.date": "2012-04-22T03:22:05.432")
```

keys()

Iterates over all of the schema keys in a flat way.

Each result of the iterator is a *key*. Each *key* is a dot-separated name. For example, the schema element *meta.observation.date* will end up in the result as the string *"meta.observation.date"*.

on_init(*init*)

Hook used to customize model attributes at the end of `__init__`.

Parameters

init
[object] First argument to `__init__`.

on_save(*path=None*)

This is a hook that is called just before saving the file. It can be used, for example, to update values in the metadata that are based on the content of the data.

Override it in the subclass to make it do something, but don't forget to "chain up" to the base class, since it does things there, too.

Parameters

path
[str] The path to the file that we're about to save to.

static open_asdf(*init=None*, *ignore_version_mismatch=True*, *ignore_unrecognized_tag=False*, ***kwargs*)

Open an asdf object from a filename or create a new asdf object

read(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

save(*path*, *dir_path=None*, **args*, ***kwargs*)

Save to either a FITS or ASDF file, depending on the path.

Parameters

path
[string or func] File path to save to. If function, it takes one argument with is model.meta.filename and returns the full path string.

dir_path: string
Directory to save to. If not None, this will override any directory information in the *path*

Returns

output_path: str
The file path the model was saved in.

search(**args*, ***kwargs*)

Search this file's tree.

Parameters

key
[NotSet, str, or any other object] Search query that selects nodes by dict key or list index. If NotSet, the node key is unconstrained. If str, the input is searched among keys/indexes as a regular expression pattern. If any other object, node's key or index must equal the queried key.

type_
[NotSet, str, or builtins.type] Search query that selects nodes by type. If NotSet, the node type is unconstrained. If str, the input is searched among (fully qualified) node type names as a regular expression pattern. If builtins.type, the node must be an instance of the input.

value
[NotSet, str, or any other object] Search query that selects nodes by value. If NotSet, the node value is unconstrained. If str, the input is searched among values as a regular expression pattern. If any other object, node's value must equal the queried value.

filter_
[callable] Callable that filters nodes by arbitrary criteria. The callable accepts one or two arguments:

- the node
- the node's list index or dict key (optional)

and returns True to retain the node, or False to remove it from the search results.

Returns

asdf.search.AsdfSearchResult
the result of the search

search_schema(*substring*)

Utility function to search the metadata schema for a particular phrase.

This is intended for interactive use, and not for use within library code.

The searching is case insensitive.

Parameters

substring
[str] The substring to search for.

Returns

locations
[list of tuples]

`set_fits_wcs`(*wcs*, *hdu_name='SCI'*)

Sets the FITS WCS information on the model using the given *astropy.wcs.WCS* object.

Note that the "key" of the WCS is stored in the WCS object itself, so it can not be set as a parameter to this method.

Parameters

wcs
[*astropy.wcs.WCS* or *pywcs.WCS* object] The object containing FITS WCS information

hdu_name
[str, optional] The name of the HDU to set the WCS from. This must use named HDU's, not numerical order HDUs. To set the primary HDU, pass `'PRIMARY'`.

`to_asdf`(*init*, **args*, ***kwargs*)

Write a data model to an ASDF file.

Parameters

init
[file path or file object]

args
[tuple, list] Additional positional arguments passed to *~asdf.AsdfFile.write_to*.

kwargs
[dict] Any additional keyword arguments are passed along to *~asdf.AsdfFile.write_to*.

`to_fits`(*init*, **args*, ***kwargs*)

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

`to_flat_dict`(*include_arrays=True*)

Returns a dictionary of all of the schema items as a flat dictionary.

Each dictionary key is a dot-separated name. For example, the schema element *meta.observation.date* will end up in the dictionary as:

```
{ "meta.observation.date": "2012-04-22T03:22:05.432" }
```

`update`(*d*, *only=None*, *extra_fits=False*)

Updates this model with the metadata elements from another model.

Note: The `update` method skips a WCS object, if present.

Parameters

d
[*~jwst.datamodels.DataModel* or dictionary-like object] The model to copy the metadata elements from. Can also be a dictionary or dictionary of dictionaries or lists.

only: str, None
Update only the named hdu, e.g. `only='PRIMARY'`. Can either be a string or list of hdu names. Default is to update all the hdus.

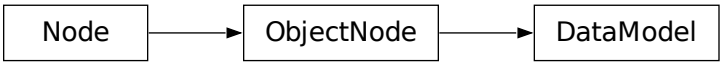
extra_fits
[boolean] Update from `extra_fits`. Default is False.

validate()
Re-validate the model instance against its schema

values()
Iterates over all of the schema values in a flat way.

write(*path*, **args*, ***kwargs*)

Class Inheritance Diagram



3.1.2 stdatamodels.asdf_in_fits Module

Functions

write(filename, tree[, hdulist])	Write ASDF data inside a fits file
open(filename_or_hdu, **kwargs)	Read ASDF data embedded in a fits file

write

stdatamodels.asdf_in_fits.**write**(*filename*, *tree*, *hdulist=None*, ***kwargs*)
Write ASDF data inside a fits file

Parameters

filename
[str or path] Filename where the resulting fits file containing the ASDF data will be saved. This is passed on to `astropy.io.fits.HDUList.writeto()`

tree
[ASDF tree or dict] ASDF data to save in the fits file

kwargs
[variable keyword arguments] Passed on to `astropy.io.fits.HDUList.writeto()`

open

stdatamodels.asdf_in_fits.**open**(*filename_or_hdu*, ***kwargs*)

Read ASDF data embedded in a fits file

Parameters

filename_or_hdu
[str, path, *astropy.io.fits.HDUList*] Filename of the fits file or an open *astropy.io.fits.HDUList* containing the ASDF data. If a filename is provided it will be opened with `astropy.io.fits.open()`.

kwargs
[variable keyword arguments] Passed on to `asdf.open()`

Returns

af
[`asdf.AsdfFile`] `asdf.AsdfFile` created from ASDF data embeded in the opened fits file.

3.1.3 stdatamodels.jwst.datamodels Package

Functions

open([init, guess, memmap])	Creates a DataModel from a number of different types

open

stdatamodels.jwst.datamodels.**open**(*init=None*, *guess=True*, *memmap=False*, ***kwargs*)

Creates a DataModel from a number of different types

Parameters

init
[shape tuple, file path, file object, astropy.io.fits.HDUList,]

numpy array, dict, None

- None: A default data model with no shape
- shape tuple: Initialize with empty data of the given shape
- file path: Initialize from the given file (FITS, JSON or ASDF)
- readable file object: Initialize from the given file object
- astropy.io.fits.HDUList: Initialize from the given *~astropy.io.fits.HDUList*
- A numpy array: A new model with the data array initialized to what was passed in.
- dict: The object model tree for the data model

guess
[bool] Guess as to the model type if the model type is not specifically known from the file. If not guess and the model type is not explicit, raise a TypeError.

memmap
[bool] Turn memmap of file on or off. (default: False).

kwargs
[dict] Additional keyword arguments passed to the DataModel constructor. Some arguments are general, others are file format-specific. Arguments of note are:

- General

  validate_arrays
  [bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

- FITS

  skip_fits_update
  [bool or None] DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Returns

model
[DataModel instance]

Classes

JwstDataModel([init, schema, memmap, ...])	**Parameters**
ABVegaOffsetModel([init])	A data model containing offsets to convert from AB to Vega magnitudes.
AmiLgModel([init, schema, memmap, ...])	A data model for AMI LG analysis results.
AmiLgFitModel([init, schema, memmap, ...])	A data model for AMI LG analysis results.
AmiOIModel([init, schema, memmap, ...])	TODO
NRMModel([init])	A data model for Non-Redundant Mask.
FgsImgApcorrModel([init])	A data model for FGS imaging apcorr reference files.
MirImgApcorrModel([init])	A data model for MIRI imaging apcorr reference files.
NrcImgApcorrModel([init])	A data model for NIRCam imaging apcorr reference files.
NisImgApcorrModel([init])	A data model for NIRISS imaging apcorr reference files.
MirLrsApcorrModel([init])	A data model for MIRI LRS apcorr reference files.
MirMrsApcorrModel([init])	A data model for MIRI MRS apcorr reference files.
NrcWfssApcorrModel([init])	A data model for NIRCam WFSS apcorr reference files.
NisWfssApcorrModel([init])	A data model for NIRISS WFSS apcorr reference files.
NrsMosApcorrModel([init])	A data model for NIRSpec MOS apcorr reference files.
NrsFsApcorrModel([init])	A data model for NIRSpec Fixed-Slit apcorr reference files.
NrsIfuApcorrModel([init])	A data model for NIRSpec IFU apcorr reference files.
AsnModel([init])	A data model for association tables.
BarshadowModel([init])	A data model for Bar Shadow correction information.
CameraModel([init, model, input_units, ...])	A model for a reference file of type "camera".
CollimatorModel([init, model, input_units, ...])	A model for a reference file of type "collimator".
CombinedSpecModel([init, schema, memmap, ...])	A data model for combined 1D spectra.
ContrastModel([init, schema, memmap, ...])	A data model for coronagraphic contrast curve files.

continues on next page

Table 2 – continued from previous page

CubeModel([init])	A data model for 3D image cubes.
DarkModel([init])	A data model for dark reference files.
DarkMIRIModel([init])	A data model for dark MIRI reference files.
DisperserModel([init, angle, gwa_tiltx, ...])	A model for a NIRSPEC reference file of type "disperser".
DistortionModel([init, model, input_units, ...])	A model for a reference file of type "distortion".
DistortionMRSModel([init, x_model, y_model, ...])	A model for a reference file of type "distortion" for the MIRI MRS.
DrizParsModel([init])	A data model for drizzle parameters reference tables.
EmiModel([init])	A data model to correct MIRI images for EMI contamination.
Extract1dImageModel([init, schema, memmap, ...])	A data model for the extract_1d reference image array.
Extract1dIFUModel([init])	A data model for IFU MIRI and NIRSpec extract 1d reference files.
FilteroffsetModel([init, filters, instrument])	A model for filter-dependent boresight offsets.
FlatModel([init])	A data model for 2D flat-field images.
NirspecFlatModel([init])	A data model for NIRSpec flat-field reference files.
NirspecQuadFlatModel([init])	A data model for NIRSpec flat-field files that differ by quadrant.
FOREModel([init, model, input_units, ...])	A model for a reference file of type "fore".
FPAModel([init, nrs1_model, nrs2_model])	A model for a NIRSPEC reference file of type "fpa".
FringeModel([init])	A data model for 2D fringe correction images.
FringeFreqModel([init])	A data model for 2D fringe correction images.
GainModel([init])	A data model for 2D gain.
GLS_RampFitModel([init, schema, memmap, ...])	A data model for the optional output of the ramp fitting step for the GLS algorithm.
GuiderRawModel([init])	A data model for Guide Star pipeline raw data files
GuiderCalModel([init])	A data model for Guide Star pipeline calibrated files
IFUCubeModel([init])	A data model for 3D IFU cubes.
NirspecIFUCubeParsModel([init])	A data model for Nirspec ifucubepars reference files.
MiriIFUCubeParsModel([init])	A data model for MIRI mrs ifucubepars reference files.
MirMrsPtCorrModel([init])	A data model for MIRI mrs IFU across-slice corrections file.
MirMrsXArtCorrModel([init])	A data model for MIRI MRS cross-artifact corrections file.
IFUFOREModel([init, model, input_units, ...])	A model for a NIRSPEC reference file of type "ifufore".
IFUImageModel([init])	A data model for 2D IFU images.
IFUPostModel([init, slice_models])	A model for a NIRSPEC reference file of type "ifupost".
IFUSlicerModel([init, model, data])	A model for a NIRSPEC reference file of type "ifuslicer".
ImageModel([init, schema, memmap, ...])	A data model for 2D images.
IPCModel([init])	A data model for IPC kernel checking information.
IRS2Model([init, schema, memmap, ...])	A data model for the IRS2 refpix reference file.
LastFrameModel([init])	A data model for Last frame correction reference files.
Level1bModel([init, schema, memmap, ...])	A data model for raw 4D ramps level-1b products.
LinearityModel([init])	A data model for linearity correction information.
MaskModel([init])	A data model for 2D masks.
MSAModel([init, models, data])	A model for a NIRSPEC reference file of type "msa".
MultiCombinedSpecModel([init])	A data model for multi-spec images.
MultiExposureModel([init])	A data model for multi-slit images derived from numerous exposures.
MultiExtract1dImageModel([init])	A data model for extract_1d reference images.
MultiSlitModel([init])	A data model for multi-slit images.

continues on next page

Table 2 – continued from previous page

MultiSpecModel([init])	A data model for multi-spec images.
NIRCAMGrismModel([init, displ, dispx, ...])	A model for a reference file of type "specwcs" for NIRCAM WFSS.
NIRISSGrismModel([init, displ, dispx, ...])	A model for a reference file of type "specwcs" for NIRISS grisms.
OTEModel([init, model, input_units, ...])	A model for a reference file of type "ote".
OutlierParsModel([init])	A data model for outlier detection parameters reference tables.
OutlierIFUOutputModel([init, schema, ...])	A data model for the optional output from outlier_detection_ifu step.
PathlossModel([init])	A data model for pathloss correction information.
MirLrsPathlossModel([init])	A data model for MIRI LRS pathloss correction information.
PersistenceSatModel([init])	A data model for the persistence saturation value (full well).
PixelAreaModel([init])	A data model for the pixel area map
NirspecSlitAreaModel([init])	A data model for the NIRSpec fixed-slit pixel area reference file
NirspecMosAreaModel([init])	A data model for the NIRSpec MOS pixel area reference file
NirspecIfuAreaModel([init])	A data model for the NIRSpec IFU pixel area reference file
FgsImgPhotomModel([init])	A data model for FGS photom reference files.
MirImgPhotomModel([init])	A data model for MIRI imaging photom reference files.
MirLrsPhotomModel([init])	A data model for MIRI LRS photom reference files.
MirMrsPhotomModel([init])	A data model for MIRI MRS photom reference files.
NrcImgPhotomModel([init])	A data model for NIRCam imaging photom reference files.
NrcWfssPhotomModel([init])	A data model for NIRCam WFSS photom reference files.
NisImgPhotomModel([init])	A data model for NIRISS imaging photom reference files.
NisSossPhotomModel([init])	A data model for NIRISS SOSS photom reference files.
NisWfssPhotomModel([init])	A data model for NIRISS WFSS photom reference files.
NrsFsPhotomModel([init])	A data model for NIRSpec Fixed-Slit photom reference files.
NrsMosPhotomModel([init])	A data model for NIRSpec MOS and IFU photom reference files.
PsfMaskModel([init])	A data model for coronagraphic 2D PSF mask reference files
QuadModel([init])	A data model for 4D image arrays.
RampModel([init])	A data model for 4D ramps.
RampFitOutputModel([init, schema, memmap, ...])	A data model for the optional output of the ramp fitting step.
ReadnoiseModel([init])	A data model for 2D readnoise.
ReferenceFileModel([init])	A data model for reference tables
ReferenceCubeModel([init])	A data model for 3D reference images
ReferenceImageModel([init])	A data model for 2D reference images.
ReferenceQuadModel([init])	A data model for 4D reference images
RegionsModel([init, regions])	A model for a reference file of type "regions".
ResetModel([init])	A data model for reset correction reference files.
ResolutionModel([init])	A data model for Spectral Resolution parameters reference tables.

continues on next page

Table 2 – continued from previous page

MiriResolutionModel([init])	A data model for MIRI Resolution reference files.
RSCDModel([init])	A data model for the RSCD reference file.
SaturationModel([init])	A data model for saturation checking information.
SlitDataModel([init])	A data model for 2D slit images.
SlitModel([init])	A data model for 2D images.
SpecModel([init, schema, memmap, ...])	A data model for 1D spectra.
SegmentationMapModel([init, schema, memmap, ...])	A data model for 2D segmentation maps
SossExtractModel([init, schema, memmap, ...])	A data model to hold NIRISS SOSS extraction model arrays.
SossWaveGridModel([init, schema, memmap, ...])	A data model to hold NIRISS SOSS wavelength grids.
SpecKernelModel([init])	A data model for 2D spectral kernels.
SpecProfileModel([init])	A data model for NIRISS SOSS spectral profile reference files.
SpecProfileSingleModel([init])	A data model for NIRISS SOSS spectral profile data.
SpecTraceModel([init])	A data model for NIRISS SOSS spectral trace reference files.
SpecTraceSingleModel([init])	A data model for NIRISS SOSS spectral trace data.
SpecwcsModel([init, model, input_units, ...])	A model for a reference file of type "specwcs".
StrayLightModel([init])	A data model for 2D straylight mask.
SuperBiasModel([init])	A data model for 2D super-bias images.
ThroughputModel([init])	A data model for filter throughput.
TrapDensityModel([init])	A data model for the trap density of a detector, for persistence.
TrapParsModel([init])	A data model for trap capture and decay parameters.
TrapsFilledModel([init, schema, memmap, ...])	A data model for the number of traps filled for a detector, for persistence.
TsoPhotModel([init, radii])	A model for a reference file of type "tsophot".
WavelengthrangeModel([init, ...])	A model for a reference file of type "wavelengthrange".
WaveCorrModel([init, apertures])	**Parameters**
WaveMapModel([init])	A data model for NIRISS SOSS wavelength map reference files.
WaveMapSingleModel([init])	A data model for NIRISS SOSS wavelength map data.
WfssBkgModel([init])	A data model for 2D WFSS master background reference files.

JwstDataModel

class stdatamodels.jwst.datamodels.**JwstDataModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: DataModel

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

crds_observatory	Get CRDS observatory code for this model.
schema_url	The schema URI to validate the model against.

Methods Summary

get_crds_parameters()	Get parameters used by CRDS to select references for this model.
on_init(init)	Hook invoked by the base class before returning a newly created model instance.
on_save(init)	Hook invoked by the base class before writing a model to a file (FITS or ASDF).

Attributes Documentation

crds_observatory

Get CRDS observatory code for this model.

Returns

str

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/core.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

get_crds_parameters()

Get parameters used by CRDS to select references for this model.

Returns

dict

on_init(*init*)

Hook invoked by the base class before returning a newly created model instance.

on_save(*init*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

ABVegaOffsetModel

class stdatamodels.jwst.datamodels.**ABVegaOffsetModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model containing offsets to convert from AB to Vega magnitudes.

Parameters

abvega_offset
[*astropy.table.Table*] An astropy table containing offsets to convert from AB to Vega magnitudes. The `abvega_offset` column represents m_AB - m_Vega.

There are three types of tables, depending on the instrument, each with different column selectors. The columns names and data types are:

- FGS
  - detector: str
  - abvega_offset: float32
- NIRCam and NIRISS
  - filter: str
  - pupil: str
  - abvega_offset: float32
- MIRI
  - filter: str
  - abvega_offset: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Methods Summary

validate()	Convenience function to be run when files are created.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/abvegaoffset.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

AmiLgModel

class stdatamodels.jwst.datamodels.**AmiLgModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for AMI LG analysis results.

Parameters

fit_image
[numpy float32 array] Fitted image

resid_image
[numpy float32 array] Residual image

closure_amp_table
[numpy table] Closure amplitudes table

closure_phase_table
[numpy table] Closure phases table

fringe_amp_table
[numpy table] Fringe amplitudes table

fringe_phase_table
[numpy table] Fringe phases table

pupil_phase_table
[numpy table] Pupil phases table

solns_table
[numpy table] Solutions table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.

- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Methods Summary

get_primary_array_name()	Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/amilg.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

get_primary_array_name()

Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created. This is intended to be overridden in the subclasses if the primary array's name is not "data".

AmiLgFitModel

class stdatamodels.jwst.datamodels.**AmiLgFitModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for AMI LG analysis results.

Parameters

centered_image
[numpy float32 array] Centered image

norm_centered_image
[numpy float32 array] Centered image normalized by data peak

fit_image
[numpy float32 array] Fitted image

norm_fit_image
[numpy float32 array] Fitted image normalized by data peak

resid_image
[numpy float32 array] Residual image

norm_resid_image
[numpy float32 array] Residual image normalized by data peak

solns_table
[numpy table] Solutions table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Methods Summary

get_primary_array_name()	Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/amilgfitmodel.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

get_primary_array_name()

Returns the name “primary” array for this model, which controls the size of other arrays that are implicitly created. This is intended to be overridden in the subclasses if the primary array’s name is not “data”.

AmiOlModel

class stdatamodels.jwst.datamodels.**AmiOIModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

TODO

Parameters

TODO

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Methods Summary

get_primary_array_name()	Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created.
on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
validate()	Re-validate the model instance against its schema

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/amioi.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

get_primary_array_name()

Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created. This is intended to be overridden in the subclasses if the primary array's name is not "data".

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

validate()

Re-validate the model instance against its schema

NRMModel

class stdatamodels.jwst.datamodels.**NRMModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for Non-Redundant Mask.

Parameters

nrm
[numpy float32 array] Non-Redundant Mask

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrm.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

FgsImgApcorrModel

class stdatamodels.jwst.datamodels.**FgsImgApcorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for FGS imaging apcorr reference files.

Parameters

apcorr_table
[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- eefraction: float32
- radius: float32
- apcorr: float32
- skyin: float32
- skyout: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.

- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/fgsimg_apcorr.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirImgApcorrModel

class stdatamodels.jwst.datamodels.**MirImgApcorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for MIRI imaging apcorr reference files.

Parameters

apcorr_table

[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- filter: str[12]
- subarray: str[15]
- eefraction: float32
- radius: float32
- apcorr: float32
- skyin: float32
- skyout: float32

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/mirimg_apcorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrcImgApcorrModel

class stdatamodels.jwst.datamodels.**NrcImgApcorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRCam imaging apcorr reference files.

Parameters

apcorr_table
[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- filter: str[12]
- pupil: str[15]
- eefraction: float32
- radius: float32
- apcorr: float32
- skyin: float32
- skyout: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrcimg_apcorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NisImgApcorrModel

class stdatamodels.jwst.datamodels.**NisImgApcorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRISS imaging apcorr reference files.

Parameters

apcorr_table
[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- filter: str[12]
- pupil: str[15]
- eefraction: float32
- radius: float32
- apcorr: float32
- skyin: float32
- skyout: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nisimg_apcorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirLrsApcorrModel

class stdatamodels.jwst.datamodels.**MirLrsApcorrModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for MIRI LRS apcorr reference files.

Parameters

apcorr_table
[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- subarray: str[15]
- wavelength: float32 1D array
- nelem_wl: int16
- size: uint8 1D array
- nelem_size: int16
- apcorr: float32 2D array
- apcorr_err: float32 2D array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/mirlrs_apcorr.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirMrsApcorrModel

class stdatamodels.jwst.datamodels.**MirMrsApcorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for MIRI MRS apcorr reference files.

Parameters

apcorr_table

[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- wavelength: float32 1D array
- radius: float32 2D array
- apcorr: float32 2D array
- apcorr_err: float32 2D array

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/mirmrs_apcorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrcWfssApcorrModel

class stdatamodels.jwst.datamodels.**NrcWfssApcorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRCam WFSS apcorr reference files.

Parameters

apcorr_table
[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- filter: str[12]
- pupil: str[15]
- wavelength: float32 1D array
- nelem_wl: int16
- size: uint8 1D array
- nelem_size: int16
- apcorr: float32 2D array
- apcorr_err: float32 2D array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrcwfss_apcorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NisWfssApcorrModel

class stdatamodels.jwst.datamodels.**NisWfssApcorrModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for NIRISS WFSS apcorr reference files.

Parameters

apcorr_table
[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- filter: str[12]
- pupil: str[15]
- wavelength: float32 1D array
- nelem_wl: int16
- size: uint8 1D array
- nelem_size: int16
- apcorr: float32 2D array
- apcorr_err: float32 2D array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors

generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/niswfss_apcorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrsMosApcorrModel

class stdatamodels.jwst.datamodels.**NrsMosApcorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRSpec MOS apcorr reference files.

Parameters

apcorr_table
[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- filter: str[12]
- grating: str[15]
- wavelength: float32 1D array
- nelem_wl: int16
- size: float32 2D array

- nelem_size: int16
- pixphase: float32 1D array
- apcorr: float32 3D array
- apcorr_err: float32 3D array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrsmos_apcorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrsFsApcorrModel

class stdatamodels.jwst.datamodels.**NrsFsApcorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRSpec Fixed-Slit apcorr reference files.

Parameters

apcorr_table
[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- filter: str[12]
- grating: str[15]
- slit: str[15]
- wavelength: float32 1D array
- nelem_wl: int16
- size: float32 2D array
- nelem_size: int16
- pixphase: float32 1D array
- apcorr: float32 3D array
- apcorr_err: float32 3D array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrsfs_apcorr.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrsIfuApcorrModel

class stdatamodels.jwst.datamodels.**NrsIfuApcorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRSpec IFU apcorr reference files.

Parameters

apcorr_table

[numpy table] Aperture correction factors table A table-like object containing row selection criteria made up of instrument mode parameters and aperture correction factors associated with those modes.

- filter: str[12]
- grating: str[15]
- wavelength: float32 1D array
- radius: float32 3D array
- apcorr: float32 3D array
- apcorr_err: float32 3D array

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrsifu_apcorr.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

AsnModel

class stdatamodels.jwst.datamodels.**AsnModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for association tables.

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values

[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation

[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment

[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays

[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.
supported_formats	

Methods Summary

parse_table()	

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/asn.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

supported_formats = ['yaml', 'json', 'fits']

Methods Documentation

parse_table()

BarshadowModel

class stdatamodels.jwst.datamodels.**BarshadowModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for Bar Shadow correction information.

Parameters

data1x1
[numpy float32 array] Bar Shadow 1x1 data array

var1x1
[numpy float32 array] Bar Shadow 1x1 correction variance

data1x3
[numpy float32 array] Bar Shadow 1x3 data array

var1x3
[numpy float32 array] Bar Shadow 1x3 correction variance

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/barshadow.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

CameraModel

class stdatamodels.jwst.datamodels.**CameraModel**(*init=None*, *model=None*, *input_units=None*, *output_units=None*, ***kwargs*)

Bases: _SimpleModel

A model for a reference file of type "camera".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

populate_meta()	Subclasses can overwrite this to populate specific meta keywords.

Attributes Documentation

reftype = 'camera'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/camera.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

populate_meta()

Subclasses can overwrite this to populate specific meta keywords.

CollimatorModel

class stdatamodels.jwst.datamodels.**CollimatorModel**(*init=None*, *model=None*, *input_units=None*, *output_units=None*, ***kwargs*)

Bases: _SimpleModel

A model for a reference file of type "collimator".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

populate_meta()	Subclasses can overwrite this to populate specific meta keywords.

Attributes Documentation

reftype = 'collimator'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/collimator.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

populate_meta()
Subclasses can overwrite this to populate specific meta keywords.

CombinedSpecModel

class stdatamodels.jwst.datamodels.**CombinedSpecModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for combined 1D spectra.

Parameters

spec_table
[numpy table] Combined, extracted spectral data table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*,

they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/combinedspec.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

ContrastModel

class stdatamodels.jwst.datamodels.**ContrastModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for coronagraphic contrast curve files.

Parameters

contrast_table
[numpy table] Contrast curve table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/contrast.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

CubeModel

class stdatamodels.jwst.datamodels.**CubeModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 3D image cubes.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

zeroframe
[numpy float32 array] Zero frame array

area
[numpy float32 array] Pixel area map array

int_times
[numpy table] table of times for each integration

wavelength
[numpy float32 array] Wavelength array

var_poisson
[numpy float32 array] Integration-specific variances of slope due to Poisson noise

var_rnoise
[numpy float32 array] Integration-specific variances of slope due to read noise

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/cube.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

DarkModel

class stdatamodels.jwst.datamodels.**DarkModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for dark reference files.

Parameters

data

[numpy float32 array] Dark current array

dq

[numpy uint32 array] 2-D data quality array for all planes

err

[numpy float32 array] Error array

dq_def

[numpy table] DQ flag definitions

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/dark.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

DarkMIRIModel

class stdatamodels.jwst.datamodels.**DarkMIRIModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for dark MIRI reference files.

Parameters

data
[numpy float32 array] Dark current array

dq
[numpy uint32 array] 2-D data quality array for all planes

err
[numpy float32 array] Error array

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/darkMIRI.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

DisperserModel

class stdatamodels.jwst.datamodels.**DisperserModel**(*init=None*, *angle=None*, *gwa_tiltx=None*, *gwa_tilty=None*, *kcoef=None*, *lcoef=None*, *tcoef=None*, *pref=None*, *tref=None*, *theta_x=None*, *theta_y=None*, *theta_z=None*, *groovedensity=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a NIRSPEC reference file of type "disperser".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'disperser'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/disperser.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

DistortionModel

class stdatamodels.jwst.datamodels.**DistortionModel**(*init=None*, *model=None*, *input_units=None*, *output_units=None*, ***kwargs*)

Bases: `_SimpleModel`

A model for a reference file of type "distortion".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'distortion'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/distortion.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

validate()
Convenience function to be run when files are created. Checks that required reference file keywords are set.

DistortionMRSModel

class stdatamodels.jwst.datamodels.**DistortionMRSModel**(*init=None*, *x_model=None*, *y_model=None*, *alpha_model=None*, *beta_model=None*, *bzero=None*, *bdel=None*, *input_units=None*, *output_units=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a reference file of type "distortion" for the MIRI MRS.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'distortion'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/distortion_mrs.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

DrizParsModel

class stdatamodels.jwst.datamodels.**DrizParsModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for drizzle parameters reference tables.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/drizpars.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

EmiModel

class stdatamodels.jwst.datamodels.**EmiModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model to correct MIRI images for EMI contamination.

Parameters

data
[numpy table] The reference waves to correct for MIRI EMI. A table-like object containing phase amplitude values corresponding to the appropriate frequency - Hz390: float32 1D array - Hz218a: float32 1D array - Hz218b: float32 1D array - Hz218c: float32 1D array - Hz164: float32 1D array - Hz10: float32 1D array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'emicorr'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/emi.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

Extract1dImageModel

class stdatamodels.jwst.datamodels.**Extract1dImageModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for the extract_1d reference image array.

Parameters

data

[numpy float32 array] 1-D extraction regions array

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.

- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/extract1dimage.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Extract1dIFUModel

class stdatamodels.jwst.datamodels.**Extract1dIFUModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for IFU MIRI and NIRSpec extract 1d reference files.

Parameters

extract1d_params

[numpy table] Basic extract 1D parameters - region_type: ascii - subtract_background: bool - method: ascii - subpixels: int16

extract1d_table

[numpy table] extract1d parameters for varying wavelengths A table-like object containing extract 1d parameters based on wavelength - wavelength: float32 1D array - radius: float32 1D array - inner_bkg: float32 1D array - outer_bkg: float32 1D array - axis_ratio: float32 1D array - axis_pa: float32 1D array

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/extract1difu.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

FilteroffsetModel

class stdatamodels.jwst.datamodels.**FilteroffsetModel**(*init=None*, *filters=None*, *instrument=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for filter-dependent boresight offsets.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

populate_meta()	
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'filteroffset'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/filteroffset.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

populate_meta()

validate()
Convenience function to be run when files are created. Checks that required reference file keywords are set.

FlatModel

class stdatamodels.jwst.datamodels.**FlatModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D flat-field images.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/flat.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NirspecFlatModel

class stdatamodels.jwst.datamodels.**NirspecFlatModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRSpec flat-field reference files.

Parameters

data
[numpy float32 array] NIRSpec flat-field reference data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error estimate

wavelength
[numpy table] Table of wavelengths for image planes

flat_table
[numpy table] Table for quickly varying component of flat field

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors

generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nirspec_flat.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NirspecQuadFlatModel

class stdatamodels.jwst.datamodels.**NirspecQuadFlatModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for NIRSpec flat-field files that differ by quadrant.

Parameters

quadrants.items.data : numpy float32 array

quadrants.items.dq : numpy uint32 array

quadrants.items.err : numpy float32 array

quadrants.items.wavelength
[numpy table] Table of wavelengths for image planes

quadrants.items.flat_table
[numpy table] Table for quickly varying component of flat field

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nirspec_quad_flat.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

FOREModel

class stdatamodels.jwst.datamodels.**FOREModel**(*init=None*, *model=None*, *input_units=None*, *output_units=None*, ***kwargs*)

Bases: _SimpleModel

A model for a reference file of type "fore".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	Subclasses can overwrite this to populate specific meta keywords.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'fore'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/fore.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

Subclasses can overwrite this to populate specific meta keywords.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

FPAModel

class stdatamodels.jwst.datamodels.**FPAModel**(*init=None*, *nrs1_model=None*, *nrs2_model=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a NIRSPEC reference file of type "fpa".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'fpa'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/fpa.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

FringeModel

class stdatamodels.jwst.datamodels.**FringeModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D fringe correction images.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/fringe.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

FringeFreqModel

class stdatamodels.jwst.datamodels.**FringeFreqModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D fringe correction images.

Parameters

rfc_freq_short : numpy table rfc_freq_medium : numpy table rfc_freq_long : numpy table max_amp : numpy table

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values

[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/fringe_freq.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

GainModel

class stdatamodels.jwst.datamodels.**GainModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D gain.

Parameters

data
[numpy float32 array] The gain

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/gain.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

GLS_RampFitModel

class stdatamodels.jwst.datamodels.**GLS_RampFitModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for the optional output of the ramp fitting step for the GLS algorithm.

Parameters

yint
[numpy float32 array] Y-intercept

sigyint
[numpy float32 array] Sigma for Y-intercept

pedestal
[numpy float32 array] Pedestal

crmag
[numpy float32 array] CR magnitudes

sigcrmag
[numpy float32 array] Sigma for CR magnitudes

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/gls_rampfit.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

GuiderRawModel

class stdatamodels.jwst.datamodels.**GuiderRawModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for Guide Star pipeline raw data files

Parameters

data
[numpy float32 array] The science data

err
[numpy float32 array] Error array

dq
[numpy uint32 array] Data quality array

planned_star_table
[numpy table] Planned reference star table

flight_star_table
[numpy table] Flight reference star table

pointing_table
[numpy table] Pointing table

centroid_table
[numpy table] Centroid packet table

track_sub_table
[numpy table] Track subarray data table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object

- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/guider_raw.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

GuiderCalModel

class stdatamodels.jwst.datamodels.**GuiderCalModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for Guide Star pipeline calibrated files

Parameters

data
[numpy float32 array] The science data

err
[numpy float32 array] Error array

dq
[numpy uint32 array] Data quality array

planned_star_table
[numpy table] Planned reference star table

flight_star_table
[numpy table] Flight reference star table

pointing_table
[numpy table] Pointing table

centroid_table
[numpy table] Centroid packet table

track_sub_table
[numpy table] Track subarray data table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object

- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/guider_cal.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

IFUCubeModel

class stdatamodels.jwst.datamodels.**IFUCubeModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 3D IFU cubes.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

weightmap
[numpy float32 array] Weight map of coverage

wavetable
[numpy table] Wavelength value for slices

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/ifucube.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NirspecIFUCubeParsModel

class stdatamodels.jwst.datamodels.**NirspecIFUCubeParsModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for Nirspec ifucubepars reference files.

Parameters

ifucubepars_table
[numpy table] default IFU cube parameters table

ifucubepars_msm_table
[numpy table] default IFU cube msm parameters table

ifucubepars_prism_msm_wavetable
[numpy table] default IFU cube prism msm wavetable

ifucubepars_med_msm_wavetable
[numpy table] default IFU cube med resolution msm wavetable

ifucubepars_high_msm_wavetable
[numpy table] default IFU cube high resolution msm wavetable

ifucubepars_emsm_table
[numpy table] default IFU cube emsm parameters table

ifucubepars_prism_emsm_wavetable
[numpy table] default IFU cube prism emsm wavetable

ifucubepars_med_emsm_wavetable
[numpy table] default IFU cube med resolution emsm wavetable

ifucubepars_high_emsm_wavetable
[numpy table] default IFU cube high resolution emsm wavetable

ifucubepars_prism_driz_wavetable
[numpy float32 array] default IFU cube prism drizzle wavetable

ifucubepars_med_driz_wavetable
[numpy float32 array] default IFU cube med resolution drizzle wavetable

ifucubepars_high_driz_wavetable
[numpy float32 array] default IFU cube high resolution drizzle wavetable

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nirspec_ifucubepars.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MiriIFUCubeParsModel

class stdatamodels.jwst.datamodels.**MiriIFUCubeParsModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for MIRI mrs ifucubepars reference files.

Parameters

ifucubepars_table
[numpy table] default IFU cube parameters table

ifucubepars_msm_table
[numpy table] default IFU cube msm parameters table

ifucubepars_multichannel_msm_wavetable
[numpy table] default IFU cube msm wavetable

ifucubepars_emsm_table
[numpy table] default IFU cube emsm parameters table

ifucubepars_multichannel_emsm_wavetable
[numpy table] default IFU cube emsm wavetable

ifucubepars_multichannel_driz_wavetable
[numpy float32 array] default IFU cube driz wavetable

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/miri_ifucubepars.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirMrsPtCorrModel

class stdatamodels.jwst.datamodels.**MirMrsPtCorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for MIRI mrs IFU across-slice corrections file.

Parameters

init
[any] Any of the initializers supported by *~jwst.datamodels.DataModel*.

leakcor_table
[numpy table] IFU spectral leak correction (fractional, Jy to Jy)

tracor_table
[numpy table] IFU across slice transmission correction

wavcorr_optical_table
[numpy table] IFU across slice wavelength offset table 1

wavcorr_xslice_table
[numpy table] IFU across slice wavelength offset table 2

wavcorr_shift_table
[numpy table] IFU across slice wavelength offset table 3

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/miri_mrsptcorr.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirMrsXArtCorrModel

class stdatamodels.jwst.datamodels.**MirMrsXArtCorrModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for MIRI MRS cross-artifact corrections file.

Parameters

init
[any] Any of the initializers supported by *~jwst.datamodels.DataModel*.

ch1a_table
[numpy table] Cross artifact correction parameters for Channel 1A

ch1b_table
[numpy table] Cross artifact correction parameters for Channel 1B

ch1c_table
[numpy table] Cross artifact correction parameters for Channel 1C

ch2a_table
[numpy table] Cross artifact correction parameters for Channel 2A

ch2b_table
[numpy table] Cross artifact correction parameters for Channel 2B

ch2c_table
[numpy table] Cross artifact correction parameters for Channel 2C

ch3a_table
[numpy table] Cross artifact correction parameters for Channel 3A

ch3b_table
[numpy table] Cross artifact correction parameters for Channel 3B

ch3c_table
[numpy table] Cross artifact correction parameters for Channel 3C

ch4a_table
[numpy table] Cross artifact correction parameters for Channel 4A

ch4b_table
[numpy table] Cross artifact correction parameters for Channel 4B

ch4c_table
[numpy table] Cross artifact correction parameters for Channel 4C

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object

- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/miri_mrsxartcorr.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

IFUFOREModel

class stdatamodels.jwst.datamodels.**IFUFOREModel**(*init=None*, *model=None*, *input_units=None*, *output_units=None*, ***kwargs*)

Bases: _SimpleModel

A model for a NIRSPEC reference file of type "ifufore".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

populate_meta()	Subclasses can overwrite this to populate specific meta keywords.

Attributes Documentation

reftype = 'ifufore'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/ifufore.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

populate_meta()
Subclasses can overwrite this to populate specific meta keywords.

IFUImageModel

class stdatamodels.jwst.datamodels.**IFUImageModel**(*init=None*, ***kwargs*)
Bases: *JwstDataModel*

A data model for 2D IFU images.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

zeroframe
[numpy float32 array] Zeroframe array

var_poisson
[numpy float32 array] variance due to poisson noise

var_rnoise
[numpy float32 array] variance due to read noise

wavelength
[numpy float32 array] wavelength

pathloss_point
[numpy float32 array] pathloss correction for point source

pathloss_uniform
[numpy float32 array] pathloss correction for uniform source

area
[numpy float32 array] Pixel area map array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.

- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/ifuimage.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

IFUPostModel

class stdatamodels.jwst.datamodels.**IFUPostModel**(*init=None*, *slice_models=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a NIRSPEC reference file of type "ifupost".

Parameters

init

[str] A file name.

slice_models

[dict] A dictionary with slice transforms with the following entries {"slice_N": {'linear': astropy.modeling.Model, ... 'xpoly': astropy.modeling.Model, ... 'xpoly_distortion': astropy.modeling.Model, ... 'ypoly': astropy.modeling.Model, ... 'ypoly_distortion': astropy.modeling.Model, ... }}

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values

[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation

[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'ifupost'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/ifupost.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

IFUSlicerModel

class stdatamodels.jwst.datamodels.**IFUSlicerModel**(*init=None*, *model=None*, *data=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a NIRSPEC reference file of type "ifuslicer".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'ifuslicer'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/ifuslicer.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

ImageModel

class stdatamodels.jwst.datamodels.**ImageModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 2D images.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

zeroframe
[numpy float32 array] Zeroframe array

var_poisson
[numpy float32 array] variance due to poisson noise

var_rnoise
[numpy float32 array] variance due to read noise

area
[numpy float32 array] Pixel area map array

pathloss_point
[numpy float32 array] Pathloss correction for point source

pathloss_uniform
[numpy float32 array] Pathloss correction for uniform source

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/image.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

IPCModel

class stdatamodels.jwst.datamodels.**IPCModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for IPC kernel checking information.

Parameters

data
[numpy float32 array] IPC deconvolution kernel

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/ipc.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

IRS2Model

class stdatamodels.jwst.datamodels.**IRS2Model**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for the IRS2 refpix reference file.

Parameters

irs2_table
[numpy table] Table for IRS2 refpix correction. A table with 8 columns and 2916352 (2048 * 712 * 2) rows. All values are float, but these are interpreted as alternating real and imaginary parts (real, imag, real, imag, ...) of complex values. There are four columns for ALPHA and four for BETA.

dq_table
[data quality info table] Table for identifying bad reference pixels. A table with three columns (OUTPUT, ODD_EVEN, and MASK) and eight rows.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.

- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/irs2.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

LastFrameModel

class stdatamodels.jwst.datamodels.**LastFrameModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for Last frame correction reference files.

Parameters

data

[numpy float32 array] Last Frame Correction array

dq

[numpy uint32 array] 2-D data quality array

err

[numpy float32 array] Error array

dq_def

[numpy table] DQ flag definitions

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values

[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/lastframe.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Level1bModel

class stdatamodels.jwst.datamodels.**Level1bModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for raw 4D ramps level-1b products.

Parameters

data
[numpy uint16 array] The science data

zeroframe
[numpy uint16 array] Zeroframe array

refout
[numpy uint16 array] Reference Output

group
[numpy table] group parameters table

int_times
[numpy table] table of times for each integration

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/level1b.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

LinearityModel

class stdatamodels.jwst.datamodels.**LinearityModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for linearity correction information.

Parameters

coeffs
[numpy float32 array] Linearity coefficients

dq
[numpy uint32 array] Data quality flags

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Methods Summary

get_primary_array_name()	Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/linearity.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

get_primary_array_name()

Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created. This is intended to be overridden in the subclasses if the primary array's name is not "data".

MaskModel

class stdatamodels.jwst.datamodels.**MaskModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D masks.

Parameters

dq

[numpy uint32 array] The mask

dq_def

[numpy table] DQ flag definitions

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object

- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Methods Summary

get_primary_array_name()	Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/mask.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

get_primary_array_name()

Returns the name "primary" array for this model, which controls the size of other arrays that are implicitly created. This is intended to be overridden in the subclasses if the primary array's name is not "data".

MSAModel

class stdatamodels.jwst.datamodels.**MSAModel**(*init=None*, *models=None*, *data=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a NIRSPEC reference file of type "msa".

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'msa'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/msa.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

MultiCombinedSpecModel

class stdatamodels.jwst.datamodels.**MultiCombinedSpecModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for multi-spec images.

This model has a special member *spec* that can be used to deal with an entire spectrum at a time. It behaves like a list:

```
>>> from stdatamodels.jwst.datamodels import CombinedSpecModel
>>> multispec_model = MultiCombinedSpecModel()
>>> multispec_model.spec.append(CombinedSpecModel())
>>> multispec_model.spec[0]
<CombinedSpecModel>
```

If *init* is a *CombinedSpecModel* instance, an empty *CombinedSpecModel* will be created and assigned to attribute *spec[0]*, and the *spec_table* attribute from the input *CombinedSpecModel* instance will be copied to the first element of *spec*. *CombinedSpecModel* objects can be appended to the *spec* attribute by using its *append* method.

Parameters

int_times
[numpy table] table of times for each integration

spec.items.spec_table
[numpy table] Extracted spectral data table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/multicombinedspec.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MultiExposureModel

class stdatamodels.jwst.datamodels.**MultiExposureModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for multi-slit images derived from numerous exposures. The intent is that all slits in this model are of the same source, with each slit representing a separate exposure of that source.

This model has a special member *exposures* that can be used to deal with an entire slit at a time. It behaves like a list:

```
>>> from .image import ImageModel
>>> multiexposure_model = MultiExposureModel()
>>> multiexposure_model.exposures.append(ImageModel())
>>> multiexposure_model.exposures[0]
<ImageModel>
```

Also, there is an extra attribute, *meta*. This will contain the meta attribute from the exposure from which each slit has been taken.

See the module *exp_to_source* for the initial creation of these models. This is part of the Level 3 processing of multi-objection observations.

Parameters

exposures.items.data : numpy float32 array

exposures.items.dq : numpy uint32 array

exposures.items.err : numpy float32 array

exposures.items.area : numpy float32 array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

core_schema_url	
schema_url	The schema URI to validate the model against.

Attributes Documentation

core_schema_url = 'http://stsci.edu/schemas/jwst_datamodel/core.schema'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/multiexposure.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MultiExtract1dImageModel

class stdatamodels.jwst.datamodels.**MultiExtract1dImageModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for extract_1d reference images.

This model has a special member *images* that can be used to deal with each image separately. It behaves like a list:

```
>>> from stdatamodels.jwst.datamodels import Extract1dImageModel
>>> multiextr1d_img_model = MultiExtract1dImageModel()
>>> multiextr1d_img_model.images.append(Extract1dImageModel())
>>> multiextr1d_img_model.images[0]
<Extract1dImageModelModel>
```

If *init* is a file name or an *Extract1dImageModel* instance, an empty *Extract1dImageModel* will be created and assigned to attribute *images[0]*, and the *data* attribute from the input array or *Extract1dImageModel* will be copied to the first element of *images*.

Parameters

images.items.data : numpy float32 array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/multiextract1d.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MultiSlitModel

class stdatamodels.jwst.datamodels.**MultiSlitModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for multi-slit images.

This model has a special member *slits* that can be used to deal with an entire slit at a time. It behaves like a list:

```
>>> from stdatamodels.jwst.datamodels import SlitModel
>>> multislit_model = MultiSlitModel()
>>> multislit_model.slits.append(SlitModel())
>>> multislit_model[0]
<SlitModel>
```

If `init` is a file name or an `ImageModel` or a `SlitModel``instance, an empty ``SlitModel` will be created and assigned to attribute `slits[0]`, and the *data*, `dq`, `err`, `var_rnoise`, and `var_poisson` attributes from the input file or model will be copied to the first element of `slits`.

Parameters

slits.items.data
[numpy float32 array] The science data

slits.items.dq
[numpy uint32 array] Data quality array

slits.items.err
[numpy float32 array] Error array

slits.items.var_poisson
[numpy float32 array] variance due to poisson noise

slits.items.var_rnoise
[numpy float32 array] variance due to read noise

slits.items.wavelength
[numpy float32 array] Wavelength array, corrected for zero-point

slits.items.barshadow
[numpy float32 array] Bar shadow correction

slits.items.flatfield_point
[numpy float32 array] flatfield array for point source

slits.items.flatfield_uniform
[numpy float32 array] flatfield array for uniform source

slits.items.pathloss_point
[numpy float32 array] pathloss array for point source

slits.items.pathloss_uniform
[numpy float32 array] pathloss array for uniform source

slits.items.photom_point
[numpy float32 array] photom array for point source

slits.items.photom_uniform
[numpy float32 array] photom array for uniform source

slits.items.area
[numpy float32 array] Pixel area map array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/multislit.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MultiSpecModel

class stdatamodels.jwst.datamodels.**MultiSpecModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for multi-spec images.

This model has a special member *spec* that can be used to deal with an entire spectrum at a time. It behaves like a list:

```
>>> from stdatamodels.jwst.datamodels import SpecModel
>>> multispec_model = MultiSpecModel()
>>> multispec_model.spec.append(SpecModel())
>>> multispec_model.spec[0]
<SpecModel>
```

If *init* is a *SpecModel* instance, an empty *SpecModel* will be created and assigned to attribute *spec[0]*, and the *spec_table* attribute from the input *SpecModel* instance will be copied to the first element of *spec*. *SpecModel* objects can be appended to the *spec* attribute by using its *append* method.

Parameters

int_times
[numpy table] table of times for each integration

spec.items.spec_table
[numpy table] Extracted spectral data table

Examples

```
>>> output_model = MultiSpecModel()
>>> spec = SpecModel()       # for the default data type
>>> for slit in input_model.slits:
...     slitname = slit.name
...     slitmodel = ExtractModel()
...     slitmodel.fromJSONFile(extref, slitname)
...     column, wavelength, countrate = slitmodel.extract(slit.data)
...     otab = np.array(zip(column, wavelength, countrate),
...                     dtype=spec.spec_table.dtype)
...     spec = datamodels.SpecModel(spec_table=otab)
...     output_model.spec.append(spec)
```

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/multispec.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NIRCAMGrismModel

class stdatamodels.jwst.datamodels.**NIRCAMGrismModel**(*init=None*, *displ=None*, *dispx=None*, *dispy=None*, *invdispl=None*, *invdispx=None*, *invdispy=None*, *orders=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a reference file of type "specwcs" for NIRCAM WFSS.

This reference file contains the models for wave, x, and y polynomial solutions that describe dispersion through the grism.

Parameters

displ: `~astropy.modeling.Model`
Nircam Grism wavelength dispersion model

dispx
[*~astropy.modeling.Model*] Nircam Grism row dispersion model

dispy
[*~astropy.modeling.Model*] Nircam Grism column dispersion model

invdispl
[*~astropy.modeling.Model*] Nircam Grism inverse wavelength dispersion model

invdispx
[*~astropy.modeling.Model*] Nircam Grism inverse row dispersion model

invdispy
[*~astropy.modeling.Model*] Nircam Grism inverse column dispersion model

orders
[*~astropy.modeling.Model*] NIRCAM Grism orders, matched to the array locations of the dispersion models

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set.

If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'specwcs'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/specwcs_nircam_grism.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

`populate_meta()`

`to_fits()`

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

`validate()`

Convenience function to be run when files are created. Checks that required reference file keywords are set.

NIRISSGrismModel

`class stdatamodels.jwst.datamodels.NIRISSGrismModel(`*init=None*, *displ=None*, *dispx=None*, *dispy=None*, *invdispl=None*, *orders=None*, *fwcpos_ref=None*, ***kwargs*`)`

Bases: *ReferenceFileModel*

A model for a reference file of type "specwcs" for NIRISS grisms.

Parameters

displ: `~astropy.modeling.Model`
NIRISS Grism wavelength dispersion model

dispx
[*~astropy.modeling.Model*] NIRISS Grism row dispersion model

dispy
[*~astropy.modeling.Model*] NIRISS Grism column dispersion model

invdispl
[*~astropy.modeling.Model*] NIRISS Grism inverse wavelength dispersion model

invdispx
[*~astropy.modeling.Model*] NIRISS Grism inverse row dispersion model

invdispy
[*~astropy.modeling.Model*] NIRISS Grism inverse column dispersion model

orders
[*~astropy.modeling.Model*] NIRISS Grism orders, matched to the array locations of the dispersion models

fwcpos_ref
[float] The reference value for the filter wheel position

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'specwcs'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/specwcs_niriss_grism.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

populate_meta()

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

OTEModel

class stdatamodels.jwst.datamodels.**OTEModel**(*init=None*, *model=None*, *input_units=None*, *output_units=None*, ***kwargs*)

Bases: `_SimpleModel`

A model for a reference file of type "ote".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

populate_meta()	Subclasses can overwrite this to populate specific meta keywords.

Attributes Documentation

reftype = 'ote'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/ote.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

populate_meta()
Subclasses can overwrite this to populate specific meta keywords.

OutlierParsModel

class stdatamodels.jwst.datamodels.**OutlierParsModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for outlier detection parameters reference tables.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/outlierpars.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

OutlierIFUOutputModel

class stdatamodels.jwst.datamodels.**OutlierIFUOutputModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for the optional output from outlier_detection_ifu step.

In the parameter definitions below, *n* is the number of exposures, *ny* and *nx* are the height and width of the image.

Parameters

diffarr
[numpy float32 array (n, ny, nx)] Minimum difference array for all the data

minarr
[numpy float32 array (ny, nx)] Final combined minimum difference array

normarr
[numpy float32 array (ny, nx)] Normalized minarr

minnorm
[numpy float32 array (ny, nx)] minarr/normarr

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.

- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/outlierifuoutput.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

PathlossModel

class stdatamodels.jwst.datamodels.**PathlossModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for pathloss correction information.

Parameters

apertures.items.pointsource_data
[numpy float32 array] Point source pathloss

apertures.items.pointsource_err
[numpy float32 array] Point source pathloss variance

apertures.items.uniform_data
[numpy float32 array] Uniform source pathloss

apertures.items.uniform_err
[numpy float32 array] Uniform source pathloss variance

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/pathloss.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirLrsPathlossModel

class stdatamodels.jwst.datamodels.**MirLrsPathlossModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for MIRI LRS pathloss correction information.

Parameters

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/mirlrs_pathloss.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

PersistenceSatModel

class stdatamodels.jwst.datamodels.**PersistenceSatModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for the persistence saturation value (full well).

Parameters

data
[numpy float32 array] Persistence saturation threshold

dq
[numpy uint32 array] data quality array

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/persat.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

PixelAreaModel

class stdatamodels.jwst.datamodels.**PixelAreaModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for the pixel area map

Parameters

data

[numpy float32 array] The pixel area array

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values

[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation

[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/pixelarea.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NirspecSlitAreaModel

class stdatamodels.jwst.datamodels.**NirspecSlitAreaModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for the NIRSpec fixed-slit pixel area reference file

Parameters

area_table
[numpy table] NIRSpec fixed-slit pixel area table A table-like object containing row selection criteria made up of the slit id and the pixel area values associated with the slits.

- slit_id: str[15]
- pixarea: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nirspec_area_slit.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NirspecMosAreaModel

class stdatamodels.jwst.datamodels.**NirspecMosAreaModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for the NIRSpec MOS pixel area reference file

Parameters

area_table
[numpy table] NIRSpec MOS pixel area table A table-like object containing row selection criteria made up of MOS shutter parameters and the pixel area values associated with the shutters.

- quadrant: int16
- shutter_x: int16
- shutter_y: int16
- pixarea: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object

- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nirspec_area_mos.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NirspecIfuAreaModel

class stdatamodels.jwst.datamodels.**NirspecIfuAreaModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for the NIRSpec IFU pixel area reference file

Parameters

area_table

[numpy table] NIRSpec IFU pixel area table A table-like object containing row selection criteria made up of IFU slice id and the pixel area values associated with the slices.

- slice_id: int16
- pixarea: float32

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nirspec_area_ifu.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

FgsImgPhotomModel

class stdatamodels.jwst.datamodels.**FgsImgPhotomModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for FGS photom reference files.

Parameters

phot_table
[numpy table] Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- photmjsr: float32
- uncertainty: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors

generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/fgsimg_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirImgPhotomModel

class stdatamodels.jwst.datamodels.**MirImgPhotomModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for MIRI imaging photom reference files.

Parameters

phot_table
[numpy table]

Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- subarray: str[15]
- photmjsr: float32

- uncertainty: float32

timecoeff
[numpy table]

Table with the coefficients for the time-dependent correction.

- amplitude: float32
- tau: float32
- t0: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/mirimg_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirLrsPhotomModel

class stdatamodels.jwst.datamodels.**MirLrsPhotomModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for MIRI LRS photom reference files.

Parameters

phot_table
[numpy table]

Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- subarray: str[15]
- photmjsr: float32
- uncertainty: float32
- nelem: int16
- wavelength: float32[*]
- relresponse: float32[*]

- reluncertainty: float32[*]

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/mirlrs_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MirMrsPhotomModel

class stdatamodels.jwst.datamodels.**MirMrsPhotomModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for MIRI MRS photom reference files.

Parameters

init
[any] Any of the initializers supported by *~jwst.datamodels.DataModel*.

data
[numpy array] An array-like object containing the pixel-by-pixel conversion values in units of (MJy / pixel) / (DN / sec).

err
[numpy array] An array-like object containing the uncertainties in the conversion values, in the same units as the data array.

dq
[numpy array] An array-like object containing bit-encoded data quality flags, indicating problem conditions for values in the data array.

dq_def
[numpy array] A table-like object containing the data quality definitions table.

pixsiz
[numpy array] An array-like object containing pixel-by-pixel size values, in units of square arcseconds (arcsec^2).

timecoeff_ch1
[numpy table] A table of time and wavelength dependent throughput corrections for channel 1

timecoeff_ch2
[numpy table] A table of time and wavelength dependent throughput corrections for channel 2

timecoeff_ch3
[numpy table] A table of time and wavelength dependent throughput corrections for channel 3

timecoeff_ch4
[numpy table] A table of time and wavelength dependent throughput corrections for channel 4

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

skip_fits_update - bool or None
DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/mirmrs_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrcImgPhotomModel

class stdatamodels.jwst.datamodels.**NrcImgPhotomModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for NIRCam imaging photom reference files.

Parameters

phot_table
[numpy table] Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- pupil: str[12]
- photmjsr: float32
- uncertainty: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)

- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrcimg_photom.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrcWfssPhotomModel

class stdatamodels.jwst.datamodels.**NrcWfssPhotomModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRCam WFSS photom reference files.

Parameters

phot_table

[numpy table] Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- pupil: str[15]
- order: int16
- photmjsr: float32
- uncertainty: float32
- nelem: int16
- wavelength: float32[*]
- relresponse: float32[*]
- reluncertainty: float32[*]

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.

- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrcwfss_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NisImgPhotomModel

class stdatamodels.jwst.datamodels.**NisImgPhotomModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for NIRISS imaging photom reference files.

Parameters

phot_table
[numpy table] Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- pupil: str[12]
- photmjsr: float32
- uncertainty: float32

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nisimg_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NisSossPhotomModel

class stdatamodels.jwst.datamodels.**NisSossPhotomModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for NIRISS SOSS photom reference files.

Parameters

phot_table
[numpy table] Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- pupil: str[15]
- order: int16
- photmj: float32
- uncertainty: float32
- nelem: int16
- wavelength: float32[*]
- relresponse: float32[*]
- reluncertainty: float32[*]

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nissoss_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NisWfssPhotomModel

class stdatamodels.jwst.datamodels.**NisWfssPhotomModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRISS WFSS photom reference files.

Parameters

phot_table
[numpy table] Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- pupil: str[15]
- order: int16
- photmjsr: float32
- uncertainty: float32
- nelem: int16
- wavelength: float32[*]
- relresponse: float32[*]
- reluncertainty: float32[*]

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/niswfss_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrsFsPhotomModel

class stdatamodels.jwst.datamodels.**NrsFsPhotomModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRSpec Fixed-Slit photom reference files.

Parameters

phot_table
[numpy table] Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- grating: str[15]
- slit: str[15]
- photmj: float32
- uncertainty: float32
- nelem: int16
- wavelength: float32[*]
- relresponse: float32[*]
- reluncertainty: float32[*]

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrsfs_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

NrsMosPhotomModel

class stdatamodels.jwst.datamodels.**NrsMosPhotomModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRSpec MOS and IFU photom reference files.

Parameters

phot_table
[numpy table] Photometric flux conversion factors table A table-like object containing row selection criteria made up of instrument mode parameters and photometric conversion factors associated with those modes.

- filter: str[12]
- grating: str[15]
- photmj: float32
- uncertainty: float32
- nelem: int16
- wavelength: float32[*]
- relresponse: float32[*]
- reluncertainty: float32[*]

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set.

If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/nrsmos_photom.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

PsfMaskModel

class stdatamodels.jwst.datamodels.**PsfMaskModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for coronagraphic 2D PSF mask reference files

Parameters

data
[numpy float32 array] The PSF mask

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/psfmask.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

QuadModel

class stdatamodels.jwst.datamodels.**QuadModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 4D image arrays.

Parameters

data

[numpy float32 array] The science data

dq

[numpy uint32 array] Data quality array

err

[numpy float32 array] Error array

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/quad.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

RampModel

class stdatamodels.jwst.datamodels.**RampModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 4D ramps.

Parameters

data
[numpy float32 array] The science data

pixeldq
[numpy uint32 array] 2-D data quality array for all planes

groupdq
[numpy uint8 array] 4-D data quality array for each plane

err
[numpy float32 array] Error array

zeroframe
[numpy float32 array] Zeroframe array

group
[numpy table] group parameters table

int_times
[numpy table] table of times for each integration

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/ramp.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

RampFitOutputModel

class stdatamodels.jwst.datamodels.**RampFitOutputModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for the optional output of the ramp fitting step.

In the parameter definitions below, *n_int* is the number of integrations, *max_seg* is the maximum number of segments that were fit, *nreads* is the number of reads in an integration, and *ny* and *nx* are the height and width of the image.

Parameters

slope
[numpy float32 array (n_int, max_seg, ny, nx)] Segment-specific slope

sigslope
[numpy float32 array (n_int, max_seg, ny, nx)] Sigma for segment-specific slope

var_poisson
[numpy float32 array (n_int, max_seg, ny, nx)] Variance due to poisson noise for segment-specific slope

var_rnoise
[numpy float32 array (n_int, max_seg, ny, nx)] Variance due to read noise for segment-specific slope

yint
[numpy float32 array (n_int, max_seg, ny, nx)] Segment-specific y-intercept

sigyint
[numpy float32 array (n_int, max_seg, ny, nx)] Sigma for segment-specific y-intercept

pedestal
[numpy float32 array (n_int, max_seg, ny, nx)] Pedestal array

weights
[numpy float32 array (n_int, max_seg, ny, nx)] Weights for segment-specific fits

crmag
[numpy float32 array (n_int, max_seg, ny, nx)] Approximate CR magnitudes

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.

- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/rampfitoutput.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

ReadnoiseModel

class stdatamodels.jwst.datamodels.**ReadnoiseModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D readnoise.

Parameters

data

[numpy float32 array] Read noise

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values

[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation

[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/readnoise.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

ReferenceFileModel

class stdatamodels.jwst.datamodels.**ReferenceFileModel**(*init=None*, ***kwargs*)
Bases: *JwstDataModel*

A data model for reference tables

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object

- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Methods Summary

print_err(message)	
save(path[, dir_path])	Save data model.
validate()	Convenience function to be run when files are created.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/referencefile.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

print_err(*message*)

save(*path*, *dir_path=None*, **args*, ***kwargs*)

Save data model. If the 'dq' and 'dq_def' exist they need special handling.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

ReferenceCubeModel

class stdatamodels.jwst.datamodels.**ReferenceCubeModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 3D reference images

Parameters

data

[numpy float32 array] The science data

dq

[numpy uint32 array] Data quality array

err

[numpy float32 array] Error array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/referencecube.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

ReferenceImageModel

class stdatamodels.jwst.datamodels.**ReferenceImageModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D reference images.

Reference image data model.

Parameters

data

[numpy float32 array] The science data

dq

[numpy uint32 array] Data quality array

err

[numpy float32 array] Error array

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/referenceimage.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

ReferenceQuadModel

class stdatamodels.jwst.datamodels.**ReferenceQuadModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 4D reference images

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors

generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/referencequad.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

RegionsModel

class stdatamodels.jwst.datamodels.**RegionsModel**(*init=None*, *regions=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a reference file of type "regions".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array

- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'regions'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/regions.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

ResetModel

class stdatamodels.jwst.datamodels.**ResetModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for reset correction reference files.

Parameters

data
[numpy float32 array] Reset Correction array

dq
[numpy uint32 array] 2-D data quality array for each integration

err
[numpy float32 array] Error array

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/reset.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

ResolutionModel

class stdatamodels.jwst.datamodels.**ResolutionModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for Spectral Resolution parameters reference tables.

Parameters

data
[numpy float32 array] Resolving Power table

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array

- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/resolution.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

MiriResolutionModel

class stdatamodels.jwst.datamodels.**MiriResolutionModel**(*init=None*, ***kwargs*)

Bases: *ResolutionModel*

A data model for MIRI Resolution reference files.

Parameters

data

[numpy table] Resolving Power table A table containing resolving power of the MRS. THe table consist of 11 columns and 12 rows. Each row corresponds to a band. The columns give the name of band, central wavelength, and polynomial coefficients (a,b,c) needed to obtain the limits and average value of the spectral resolution.

psf_fwhm_alpha_table

[table] PSF FWHM Alpha A table with 5 columns. Column 1 gives the cutoff wavelength where the polynomials describing alpha FWHM change. Columns 2 and 3 give the polynomial coefficients (a,b) describing alpha FWHM for wavelengths shorter than cutoff. Columns 4 and 5 give the polynomial coefficients (a,b) describing alpha FWHM for wavelengths longer than the cutoff.

psf_fwhm_beta_table

[table] PSF FWHM Beta A table with 5 columns. Column 1 gives the cutoff wavelength where the polynomials describing alpha FWHM change. Columns 2 and 3 give the polynomial coefficients (a,b) describing beta FWHM for wavelengths shorter than cutoff. Columns 4 and 5 give the polynomial coefficients (a,b) describing beta FWHM for wavelengths longer than the cutoff.

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/miri_resolution.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

RSCDModel

class stdatamodels.jwst.datamodels.**RSCDModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for the RSCD reference file.

Parameters

rscd_group_skip_table
[numpy table] Reference table for RSCD correction baseline correction A table with 3 columns that set the number of groups to skip for each subarray and readpatt

rscd_gen_table
[numpy table] Reference table for RSCD correction enhanced correction A table with 5 columns that sets up general parameters for the enhanced correction

rscd_int1_table
[numpy table] Reference table for RSCD correction enhanced correction A table with 7 columns that sets up correction parameters for the enhanced correction for integration 1 based on subarray, readpatt, even or odd row

rscd_int2_table
[numpy table] Reference table for RSCD correction enhanced correction A table with 7 columns that sets up correction parameters for the enhanced correction for integration 2 based on subarray, readpatt, even or odd row

rscd_int3_table
[numpy table] Reference table for RSCD correction enhanced correction A table with 7 columns that sets up correction parameters for the enhanced correction for integration 3 based on subarray, readpatt, even or odd row

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/rscd.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SaturationModel

class stdatamodels.jwst.datamodels.**SaturationModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for saturation checking information.

Parameters

data
[numpy float32 array] Saturation threshold

dq
[numpy uint32 array] 2-D data quality array for all planes

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors

generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/saturation.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SlitDataModel

class stdatamodels.jwst.datamodels.**SlitDataModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 2D slit images.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

var_poisson
[numpy float32 array] variance due to poisson noise

var_rnoise
[numpy float32 array] variance due to read noise

var_flat
[numpy float32 array] variance due to flat

wavelength
[numpy float32 array] Wavelength array, corrected for zero-point

barshadow
[numpy float32 array] Bar shadow correction

flatfield_point
[numpy float32 array] flatfield array for point source

flatfield_uniform
[numpy float32 array] flatfield array for uniform source

pathloss_point
[numpy float32 array] pathloss array for point source

pathloss_uniform
[numpy float32 array] pathloss array for uniform source

photom_point
[numpy float32 array] photom array for point source

photom_uniform
[numpy float32 array] photom array for uniform source

area
[numpy float32 array] Pixel area map array

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the meta-

data. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/slitdata.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SlitModel

class stdatamodels.jwst.datamodels.**SlitModel**(*init=None*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 2D images.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

var_poisson
[numpy float32 array] variance due to poisson noise

var_rnoise
[numpy float32 array] variance due to read noise

var_flat
[numpy float32 array] variance due to flat

wavelength
[numpy float32 array] Wavelength array, corrected for zero-point

barshadow
[numpy float32 array] Bar shadow correction

flatfield_point
[numpy float32 array] flatfield array for point source

flatfield_uniform
[numpy float32 array] flatfield array for uniform source

pathloss_point
[numpy float32 array] pathloss array for point source

pathloss_uniform
[numpy float32 array] pathloss array for uniform source

photom_point
[numpy float32 array] photom array for point source

photom_uniform
[numpy float32 array] photom array for uniform source

area
[numpy float32 array] Pixel area map array

int_times
[numpy table] table of times for each integration

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/slit.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SpecModel

class stdatamodels.jwst.datamodels.**SpecModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 1D spectra.

Parameters

spec_table

[numpy table] Extracted spectral data table A table with at least four columns: wavelength, flux, an error estimate for the flux, and data quality flags.

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/spec.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SegmentationMapModel

class stdatamodels.jwst.datamodels.**SegmentationMapModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for 2D segmentation maps

Parameters

data
[numpy uint32 array] The segmentation map

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set.

If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/segmap.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SossExtractModel

class stdatamodels.jwst.datamodels.**SossExtractModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model to hold NIRISS SOSS extraction model arrays. For each order, stores the model trace per integration and aperture pixel weights for each order extraction.

This model is written to explicitly handle each of the three orders.

Parameters

order1
[numpy float32 array] 3-D array of the 2-D model trace for each integration, for spectral order 1

order2
[numpy float32 array] 3-D array of the 2-D model trace for each integration, for spectral order 2

order3
[numpy float32 array] 3-D array of the 2-D model trace for each integration, for spectral order 3

aperture1
[numpy float32 array] 2-D array storing the pixel weights for box-extracting spectral order 1

aperture2
[numpy float32 array] 2-D array storing the pixel weights for box-extracting spectral order 2

aperture3
[numpy float32 array] 2-D array storing the pixel weights for box-extracting spectral order 3

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors

generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/sossextractmodel.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SossWaveGridModel

class stdatamodels.jwst.datamodels.**SossWaveGridModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model to hold NIRISS SOSS wavelength grids. This 1-D array of wavelengths can be saved from a processing run and applied to future input products.

Parameters

wavegrid
[numpy float32 array] 1-D array of the wavelengths corresponding to the ATOCA fit.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/sosswavegrid.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SpecKernelModel

class stdatamodels.jwst.datamodels.**SpecKernelModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D spectral kernels.

Parameters

wavelengths
[numpy float32 array] Wavelengths

kernels
[numpy float32 array] Kernel values

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/speckernel.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SpecProfileModel

class stdatamodels.jwst.datamodels.**SpecProfileModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRISS SOSS spectral profile reference files.

This model has a special member *profile* that can be used to deal with an entire spectral profile at a time. It behaves like a list:

```
>>> from stdatamodels.jwst.datamodels import SpecProfileSingleModel
>>> specprofile_model = SpecProfileModel()
>>> specprofile_model.profile.append(SpecProfileSingleModel())
>>> specprofile_model.profile[0]
<SpecProfileSingleModel>
```

If *init* is a *SpecProfileSingleModel* instance, an empty *SpecProfileSingleModel* will be created and assigned to attribute *profile[0]*, and the *data* attribute from the input *SpecProfileSingleModel* instance will be copied to the first element of *profile*. *SpecProfileSingleModel* objects can be appended to the *profile* attribute by using its *append* method.

Parameters

profile.items.data

[numpy array] Spectral profile data

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/specprofile.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SpecProfileSingleModel

class stdatamodels.jwst.datamodels.**SpecProfileSingleModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRISS SOSS spectral profile data.

Parameters

data
[numpy float32 array] Spectral profile values

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/specprofilesingle.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SpecTraceModel

class stdatamodels.jwst.datamodels.**SpecTraceModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for NIRISS SOSS spectral trace reference files.

This model has a special member *trace* that can be used to deal with an entire spectral trace at a time. It behaves like a list:

```
>>> from stdatamodels.jwst.datamodels import SpecTraceSingleModel
>>> spectrace_model = SpecTraceModel()
>>> spectrace_model.trace.append(SpecTraceSingleModel())
>>> spectrace_model.trace[0]
<SpecTraceSingleModel>
```

If *init* is a *SpecTraceSingleModel* instance, an empty *SpecTraceSingleModel* will be created and assigned to attribute *trace[0]*, and the *data* attribute from the input *SpecTraceSingleModel* instance will be copied to the first element of *trace*. *SpecTraceSingleModel* objects can be appended to the *trace* attribute by using its *append* method.

Parameters

trace.items.data
[numpy table] Spectral trace data

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/spectrace.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SpecTraceSingleModel

class stdatamodels.jwst.datamodels.**SpecTraceSingleModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRISS SOSS spectral trace data.

Parameters

data
[numpy table] Trace values

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/spectracesingle.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SpecwcsModel

class stdatamodels.jwst.datamodels.**SpecwcsModel**(*init=None*, *model=None*, *input_units=None*, *output_units=None*, ***kwargs*)

Bases: _SimpleModel

A model for a reference file of type "specwcs".

Notes

For NIRISS and NIRCAM WFSS modes the specwcs file is used during extract_2D. See NIRCAMGrismModel and NIRISSGrismModel.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'specwcs'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/specwcs.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

validate()
Convenience function to be run when files are created. Checks that required reference file keywords are set.

StrayLightModel

class stdatamodels.jwst.datamodels.**StrayLightModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D straylight mask.

Parameters

data
[numpy uint8 array] Straylight mask

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/straylight.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

SuperBiasModel

class stdatamodels.jwst.datamodels.**SuperBiasModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for 2D super-bias images.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/superbias.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

ThroughputModel

class stdatamodels.jwst.datamodels.**ThroughputModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for filter throughput.

Parameters

filter_table

[numpy table] Filter throughput table

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values

[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/throughput.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

TrapDensityModel

class stdatamodels.jwst.datamodels.**TrapDensityModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for the trap density of a detector, for persistence.

Parameters

data
[numpy float32 array] Trap density

dq
[numpy uint32 array] data quality array

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/trapdensity.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

TrapParsModel

class stdatamodels.jwst.datamodels.**TrapParsModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for trap capture and decay parameters.

Parameters

trappars_table
[numpy table] Trap capture and decay parameters A table with three columns for trap-capture parameters and one column for the trap-decay parameter. Each row of the table is for a different trap family.

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.

- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/trappars.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

TrapsFilledModel

class stdatamodels.jwst.datamodels.**TrapsFilledModel**(*init=None*, *schema=None*, *memmap=False*, *pass_invalid_values=None*, *strict_validation=None*, *validate_on_assignment=None*, *validate_arrays=False*, *ignore_missing_extensions=True*, ***kwargs*)

Bases: *JwstDataModel*

A data model for the number of traps filled for a detector, for persistence.

Parameters

data

[numpy float32 array] Traps filled The map of the number of traps filled over the detector, with one plane for each "trap family."

Parameters

init

[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema

[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap

[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values

[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/trapsfilled.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

TsoPhotModel

class stdatamodels.jwst.datamodels.**TsoPhotModel**(*init=None*, *radii=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A model for a reference file of type "tsophot".

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.

- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'tsophot'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/tsophot.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

WavelengthrangeModel

class stdatamodels.jwst.datamodels.**WavelengthrangeModel**(*init=None*, *wrange_selector=None*, *wrange=None*, *order=None*, *extract_orders=None*, *wunits=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A model for a reference file of type "wavelengthrange".

The model is used by MIRI, NIRSPEC, NIRCAM, and NIRISS.

Parameters

wrange
[list] Contains a list of [order, filter, min wave, max wave]

order
[list] A list of orders that are available and described in the file

extract_orders
[list] A list of filters and the orders that should be extracted by default

wunits
[*~astropy.units*] The units for the wavelength data

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

get_wfss_wavelength_range(filter, orders)	Retrieve the wavelength range for a WFSS observation.
on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
to_fits()	Write a data model to a FITS file.
validate()	Convenience function to be run when files are created.

Attributes Documentation

reftype = 'wavelengthrange'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/wavelengthrange.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

get_wfss_wavelength_range(*filter*, *orders*)

Retrieve the wavelength range for a WFSS observation.

Parameters

filter
[str] Filter for which to retrieve the wavelength range.

orders
[list] List of spectral orders

Returns

wave_range
[dict] Pairs of {order: (wave_min, wave_max)} for each order and the specific filter.

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

to_fits()

Write a data model to a FITS file.

Parameters

init
[file path or file object]

args, kwargs
Any additional arguments are passed along to *astropy.io.fits.writeto*.

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

WaveCorrModel

class stdatamodels.jwst.datamodels.**WaveCorrModel**(*init=None*, *apertures=None*, ***kwargs*)

Bases: *ReferenceFileModel*

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.

- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

aperture_names	
reftype	
schema_url	The schema URI to validate the model against.

Methods Summary

on_save([path])	Hook invoked by the base class before writing a model to a file (FITS or ASDF).
populate_meta()	
validate()	Convenience function to be run when files are created.

Attributes Documentation

aperture_names

reftype = 'wavecorr'

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/wavecorr.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Methods Documentation

on_save(*path=None*)

Hook invoked by the base class before writing a model to a file (FITS or ASDF).

populate_meta()

validate()

Convenience function to be run when files are created. Checks that required reference file keywords are set.

WaveMapModel

class stdatamodels.jwst.datamodels.**WaveMapModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for NIRISS SOSS wavelength map reference files.

This model has a special member *map* that can be used to deal with an entire wavelength map at a time. It behaves like a list:

```
>>> from stdatamodels.jwst.datamodels import WaveMapSingleModel, WaveMapModel
>>> wavemap_model = WaveMapModel()
>>> wavemap_model.map.append(WaveMapSingleModel())
>>> wavemap_model.map[0]
<WaveMapSingleModel>
```

If *init* is a *WaveMapSingleModel* instance, an empty *WaveMapSingleModel* will be created and assigned to attribute *map[0]*, and the *data* attribute from the input *WaveMapSingleModel* instance will be copied to the first element of *map*. *WaveMapSingleModel* objects can be appended to the *map* attribute by using its *append* method.

Parameters

map.items.data
[numpy data array] Wavelength map data

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/wavemap.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

WaveMapSingleModel

class stdatamodels.jwst.datamodels.**WaveMapSingleModel**(*init=None*, ***kwargs*)
Bases: *ReferenceFileModel*

A data model for NIRISS SOSS wavelength map data.

Parameters

data
[numpy float32 array] Wavelength values

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object

- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/wavemapsingle.schema'

The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

WfssBkgModel

class stdatamodels.jwst.datamodels.**WfssBkgModel**(*init=None*, ***kwargs*)

Bases: *ReferenceFileModel*

A data model for 2D WFSS master background reference files.

Parameters

data
[numpy float32 array] The science data

dq
[numpy uint32 array] Data quality array

err
[numpy float32 array] Error array

dq_def
[numpy table] DQ flag definitions

Parameters

init
[str, tuple, *~astropy.io.fits.HDUList*, ndarray, dict, None]

- None : Create a default data model with no shape.
- tuple : Shape of the data array. Initialize with empty data array with shape specified by the.
- file path: Initialize from the given file (FITS or ASDF)
- readable file object: Initialize from the given file object
- *~astropy.io.fits.HDUList* : Initialize from the given *~astropy.io.fits.HDUList*.
- A numpy array: Used to initialize the data array
- dict: The object model tree for the data model

schema
[dict, str (optional)] Tree of objects representing a JSON schema, or string naming a schema. The schema to use to understand the elements on the model. If not provided, the schema associated with this class will be used.

memmap
[bool] Turn memmap of FITS/ASDF file on or off. (default: False).

pass_invalid_values
[bool or None] If *True*, values that do not validate the schema will be added to the metadata. If *False*, they will be set to *None*. If *None*, value will be taken from the environmental PASS_INVALID_VALUES. Otherwise the default value is *False*.

strict_validation
[bool or None] If *True*, schema validation errors will generate an exception. If *False*, they will generate a warning. If *None*, value will be taken from the environmental STRICT_VALIDATION. Otherwise, the default value is *False*.

validate_on_assignment
[bool or None] Defaults to 'None'. If *None*, value will be taken from the environmental VALIDATE_ON_ASSIGNMENT, defaulting to 'True' if no environment variable is set. If 'True', attribute assignments are validated at the time of assignment. Validation errors generate warnings and values will be set to *None*. If 'False', schema validation occurs only once at the time of write. Validation errors generate warnings.

validate_arrays
[bool] If *True*, arrays will be validated against ndim, max_ndim, and datatype validators in the schemas.

ignore_missing_extensions
[bool] When *False*, raise warnings when a file is read that contains metadata about extensions that are not available. Defaults to *True*.

kwargs
[dict] Additional keyword arguments passed to lower level functions. These arguments are generally file format-specific. Arguments of note are:

- FITS

  skip_fits_update - bool or None
  DEPRECATED *True* to skip updating the ASDF tree from the FITS headers, if possible. If *None*, value will be taken from the environmental SKIP_FITS_UPDATE. Otherwise, the default value is *True*.

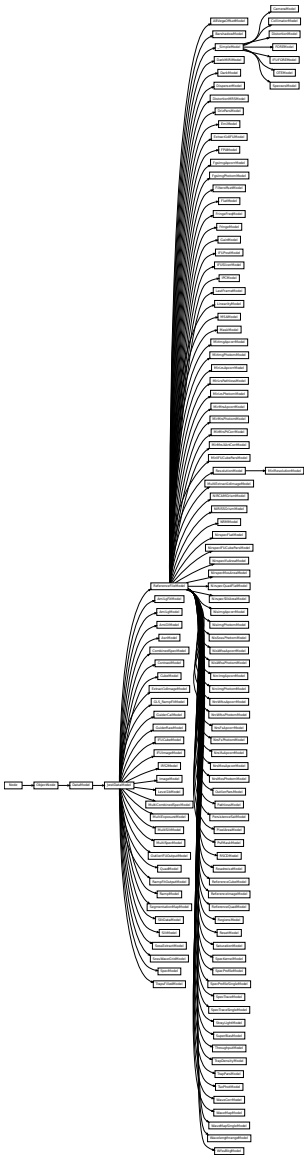
Attributes Summary

schema_url	The schema URI to validate the model against.

Attributes Documentation

schema_url = 'http://stsci.edu/schemas/jwst_datamodel/wfssbkg.schema'
The schema URI to validate the model against. If None, only basic validation of required metadata properties (filename, model_type) will occur.

Class Inheritance Diagram



3.1.4 stdatamodels.jwst.transforms Package

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

j

s

INDEX

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W