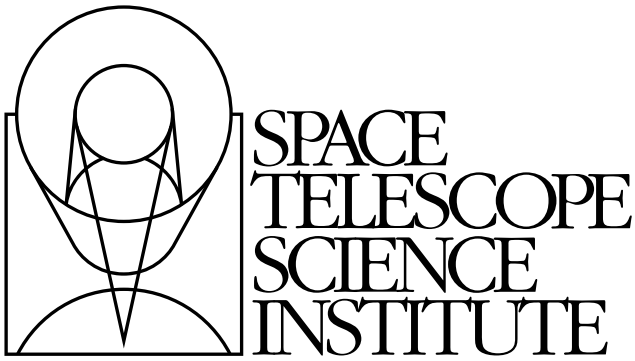

fitsblender Documentation

Release 0.1

Michael Droettboom

June 09, 2014

CONTENTS

fitsblender.**fitsblender**(*headers*, *spec*)

Given a list of FITS headers, aggregates values and creates a table made up of values from a number of headers, according to the given specification.

Parameters:

- *headers* is a sequence where each element is either:
  - a `pyfits.Header` instance
  - a 2-tuple of the form (*filename*, *extension*), where *filename* is a path to a FITS file and *extension* is an extension number.
- *spec* is a list defining which keyword arguments are to be aggregated and how. Each element in the list should be a sequence with 2 to 5 elements of the form:

  (*src_keyword*, *dst_name*, *function*, *error_type*, *error_value*)

  - *src_keyword* is the keyword to pull values from. It is case-insensitive.
  - *dst_name* is the name to use as a dictionary key or column name for the destination values.
  - *function* (optional). If function is not None, the values from the source are aggregated and returned in the *aggregate_dict*. If function is None (or the tuple contains only 2 elements), all values are stored as a column with the name *dst_name* in the result *table*.

    If not None, *function* should be a callable object that takes a sequence of values and returns an aggregate result. If the function returns None, no values will be added to the aggregate dictionary. There are many functions in Numpy that are directly useful as an aggregating function, for example:

    - mean: `numpy.mean`
    - median: `numpy.median`
    - maximum: `numpy.max`
    - minimum: `numpy.min`
    - total: `numpy.sum`
    - standard deviation: `numpy.std`

    Lambda functions are also often useful:

    - first: `lambda x: x[0]`
    - last: `lambda x: x[-1]`

    Additionally, *function* may be a tuple, where each member is itself a callable object. The result will be a tuple containing results from each of the given functions. For instance, to aggregate a range of values, i.e. both the minimum and maximum values, use the following as *function*: `(numpy.min, numpy.max)`.
  - *error_type* (optional) defines how missing or syntax-errored values are handled. It may be one of the following:
    - 'ignore': missing or unparsable values are ignored. They are not included in the list of values passed to the aggregating function. In the result *table*, missing values are masked out.
    - 'raise': missing or unparsable values raise a `ValueError` exception.
    - 'constant': missing or unparsable values are replaced with a constant, given by the *error_value* field.
  - *error_value* (optional) is the constant value to be used for missing or unparsable values when *error_type* is set to 'constant'. When not provided, it defaults to `NaN`.

Returns:

A 2-tuple of the form (*aggregate_dict*, *table*) where:

- *aggregate_dict* is a dictionary of where the keys come from *dst_name* and the values are the aggregated values as run through *function*.
- *table* is a masked Numpy structured array where the column names come from *dst_name* and the column contains the values from *src_keyword* for all of the given headers. Missing values are masked out.

CHAPTER

ONE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

f

PYTHON MODULE INDEX

f

INDEX

F