## CONTENTS

1 Primary User Interface: `calcos()`  
2 Indices and tables  
   Python Module Index  
   Python Module Index  
   Index  

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Primary User Interface: <code>calcos()</code></td>
<td>3</td>
</tr>
<tr>
<td>2 Indices and tables</td>
<td>7</td>
</tr>
<tr>
<td>Python Module Index</td>
<td>9</td>
</tr>
<tr>
<td>Python Module Index</td>
<td>11</td>
</tr>
<tr>
<td>Index</td>
<td>13</td>
</tr>
</tbody>
</table>
Contents:
PRIMARY USER INTERFACE:
CALCOS()

1. To run this task from within Python:

   ```python
   >>> import calcos
   >>> calcos.calcos("rootname_asn.fits")
   >>> calcos.calcos("rootname_rawtag_a.fits")
   >>> from stsci.tools import teal
   >>> teal.teal("calcos")
   ```

2. To run this task using the TEAL GUI to set the parameters under PyRAF:

   ```python
   >>> import calcos
   >>> teal calcos  # or ‘epar calcos’
   ```

3. To run this task from the operating system command line:

   ```bash
   # Calibrate an entire association.
   $ calcos rootname_asn.fits

   # Calibrate xyz_rawtag_a.fits (and xyz_rawtag_b.fits, if present)
   $ calcos xyz_rawtag_a.fits
   ```

   ```python
   calcos(calcos.asntable, outdir=None, verbosity=None, find_target={'cutoff': None, 'flag': False}, create_csum_image=False, raw_csum_coords=False, only_csum=False, binx=None, biny=None, compress_csum=False, compression_parameters='gzip -0.01', shift_file=None, save_temp_files=False, stimfile=None, livetimefile=None, burstfile=None)
   ```

   Calibrate COS data.

   This is the main module for calibrating COS data.

   **Parameters**

   asntable: str :
   The rootname (with “_asn”) of an association file, or the rootname (with “_raw”) of a raw file. If the value of a raw FUV file is specified and files for both segments are present, then both of those files will be calibrated (i.e. without having to explicitly list both files).

   **Returns**

   status: int :
   0 is OK; 5 means no file was found that could be calibrated.
Other Parameters

outdir: str or None, optional :
Name of output directory.

verbosity: int {0, 1, 2} or None, optional :
If not None, set verbosity to this level.

find_target: dictionary, optional :
Keys are “flag” and “cutoff”. flag = True means use the location of the target in the
cross-dispersion direction if the standard deviation (pixels) of the location is less than
or equal to cutoff (if cutoff is positive). flag = False means use the location determined
from the wavecal.

create_csum_image: boolean, optional :
If True, write an image that reflects the counts detected at each pixel (includes deadcorr
but not flatcorr), for OPUS to add to the cumulative image.

raw_csum_coords: boolean, optional :
If True, use raw pixel coordinates (rather than thermally and geometrically corrected)
to create the csum image.

only_csum: boolean, optional :
If True, create a csum image, but most other files will not be written.

binx, biny: int or None, optional :
Binning factor for the X and Y axes, or None, which means that the default binning
(currently 1) should be used.

compress_csum: boolean, optional :
If True, compress the “calcos sum” image.

compression_parameters: string, optional :
Two values separated by a comma; the first is the compression type (rice, gzip or hcom-
press), and the second is the quantization level. The default is “gzip,-0.01”.

shift_file: str, optional :
If specified, this text file contains values of shift1 (and possibly shift2) to override the
values found via wavecal processing.

save_temp_files: boolean, optional :
By default, the _x1d_a.fits and _x1d_b.fits files (if FUV) will be deleted after concate-
nating to the _x1d.fits file. Specify save_temp_files=True to keep these files.

stimfile: str, optional :
If specified, the stim positions will be written to (or appended to) a text file with this
name.

livetimefile: str, optional :
If specified, the livetime factors will be written to (or appended to) a text file with this
name.

burstfile: str, optional :
If specified, burst information will be written to (or appended to) a text file with this name.

**class calcos.Association(asntable, outdir, cl_args)**

Read and interpret the association table.

**Parameters**

- **asntable: str**:
  The rootname (with “_asn”) of an association file, or the rootname (with “_raw”) of a raw file (or pair of files if FUV).

- **outdir: str or None**:
  Name of output directory.

- **cl_args: dictionary**:
  Some of the command-line arguments, or their defaults.

**Constructor**

**class calcos.Observation(input, outdir, memtype, suffix, shift_file, first)**

Get information about an observation from its headers.

This base class is not directly used; one of its subclasses will be invoked, depending on DETECTOR and OBSMODE.

**Parameters**

- **input: str**:
  The name of an input raw file.

- **outdir: str**:
  An empty string or the name of the output directory.

- **memtype: str**:
  Read from the association table; used to distinguish between wavecal and science observation.

- **suffix: str**:
  Suffix to the rootname, but just “_rawtag” or “_rawaccum” (i.e. excluding “_a” or “_b” if the data were taken with the FUV detector); this may be reset internally to “_corrtag” or “_rawimage” or “_rawacq”.

- **shift_file: str or None**:
  The name of the shift file (command-line argument), if one was specified.

- **first: boolean**:
  True if the current file is the first of two for FUV.

Invoked by a subclass.
INDICES AND TABLES

- genindex
- modindex
- search
C
  calcos.3
PYTHON MODULE INDEX

calcos.3
INDEX

A
Association (class in calcos), 5

C
calcos (module), 3
calcos() (in module calcos), 3

O
Observation (class in calcos), 5