

wfc3tools Documentation

Release 1.0

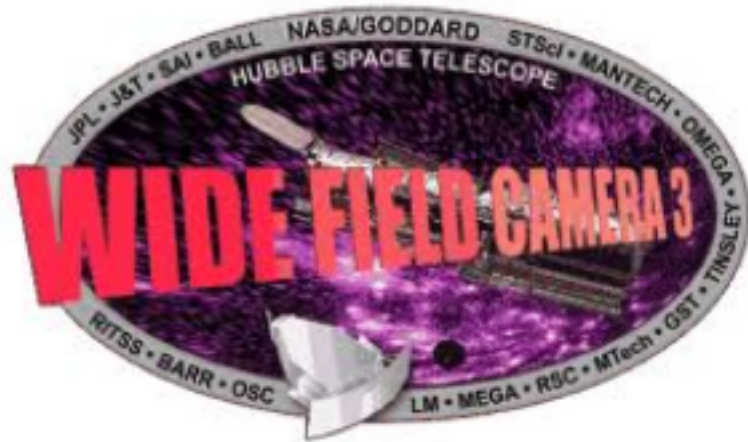
Megan Sosey

June 09, 2014

CONTENTS

1	calwf3	3
1.1	Examples	3
1.2	Running calwf3	3
1.3	Batch calwf3	3
1.4	Where to Find calwf3	4
1.5	Usage	4
1.6	Command Line Options	4
1.7	Unit Conversion to Electrons	4
1.8	Dark Current Subtraction (DARKCORR)	4
1.9	Post-Flash Correction (FLSHCORR)	5
1.10	FLATCORR	5
1.11	Photometry Keywords (PHOTCORR)	5
1.12	calwf3 Output	5
2	wf32d	7
2.1	Examples	7
2.2	Parameters	7
3	wf3ccd	9
3.1	Example	9
3.2	Parameters	9
4	wf3ir	11
4.1	Examples	11
4.2	Parameters	12
5	wf3rej	13
5.1	Example	13
5.2	Parameters	13
6	Running Astrodrizzle	15
6.1	Typical Usage	15
6.2	Alternative Usage	15
6.3	GUI Usage under Python	15
6.4	PyRAF Usage	15
6.5	Options	15
7	sampinfo	17
7.1	Options	17
7.2	Usage	17

7.3	Example Output	18
8	pstat	21
8.1	Parameters	21
8.2	Usage	22
9	pstack	23
9.1	Parameters	23
9.2	Usage	24
10	sub2full	25
10.1	USAGE	25
10.2	PARAMETERS	25
10.3	RETURNS	26
10.4	Example Output	26
10.5	More information on header keywords	26
11	embedsub	29
11.1	USAGE	29
11.2	PARAMETERS	29
11.3	RETURNS	29
11.4	Example Output	29
12	Software Update History for HSTCAL.CALWF3	31
13	Indices and tables	49



See the team website for more information: <http://www.stsci.edu/hst/wfc3/>

The current WFC3 Data Handbook can be found at <http://www.stsci.edu/hst/wfc3/documents/handbooks/currentDHB/>

.

Contents:

CALWF3

1.1 Examples

In Python without TEAL:

```
>>> from wfc3tools import calwf3
>>> calwf3.calwf3(filename)
```

In Python with TEAL:

```
>>> from stsci.tools import teal
>>> from wfc3tools import calwf3
>>> teal.teal('calwf3')
```

In Pyraf:

```
>>> import wfc3tools
>>> epar calwf3
```

A detailed description of this new and improved `calwf3` will be available in a future publication of WFC3 Data Handbook. The current WFC3 Data Handbook can be found at <http://www.stsci.edu/hst/wfc3/documents/handbooks/currentDHB/>. In the meantime, if you have questions not answered in this documentation, please contact STScI Help Desk ([help\[at\]stsci.edu](mailto:help[at]stsci.edu)).

1.2 Running calwf3

`calwf3` can be run on a single input raw file or an `asn` table listing the members of an association. When processing an association, it retrieves calibration switch and reference file keyword settings from the first image listed in the `asn` table. `calwf3` does not accept a user-defined list of input images on the command line (e.g. `*raw.fits` to process all raw files in the current directory).

The `wf3ccd`, `wf32d`, and `wf3ir` tasks on the other hand, will accept such user-defined input file lists, but they will not accept an association table (`asn`) as input.

1.3 Batch calwf3

The recommended method for running `calwf3` in batch mode is to use Python and the `wfc3tools` package in the “STSDAS distribution <http://www.stsci.edu/institute/software_hardware/stsdas/download-stsdas>.”

For example:

```
from wfc3tools import calwf3
import glob

for fits in glob.glob('j*_raw.fits'):
    calwf3.calwf3(fits)
```

1.4 Where to Find calwf3

calwf3 is now part of HSTCAL package, which can be downloaded from http://www.stsci.edu/institute/software_hardware/stsdas/download-stsdas

1.5 Usage

The calwf3 executable can also be called directly from the command line:

```
>>> calwf3.e iaa001kaq_raw.fits [command line options]
```

1.6 Command Line Options

calwf3 supports several command line options:

- -t: Print verbose time stamps.
- -s: Save temporary files.
- -v: Turn on verbose output.
- -d: Turn on debug output.
- -q: Turn on quiet output.
- -r: Print the current software version number (revision)
- --version: Print the current software version

1.7 Unit Conversion to Electrons

The UVIS image is multiplied by gain right after BIASCORR, converting it to ELECTRONS. This step is no longer embedded within FLATCORR.

1.8 Dark Current Subtraction (DARKCORR)

It uses DARKFILE for the reference dark image.

The UVIS Dark image is now scaled by EXPTIME and FLASHDUR.

1.9 Post-Flash Correction (FLSHCORR)

Post-flash correction is now performed after DARKCORR in the WF32D step. When FLSHCORR=PERFORM, it uses FLSHFILE (the post-flash reference file).

1.10 FLATCORR

Conversion from DN to ELECTRONS no longer depends on FLATCORR=PERFORM. Unit conversion is done for all exposures after BIASCORR.

1.11 Photometry Keywords (PHOTCORR)

The PHOTCORR step is now performed using tables of precomputed values instead of calls to SYNPHOT. The correct table for a given image must be specified in the IMPHTTAB header keyword in order for calwf3 to perform the PHOTCORR step. By default, it should be in the `iref` directory and have the suffix `_imp.fits`. Each DETECTOR uses a different table.

If you do not wish to use this feature, set PHOTCORR to OMIT.

1.12 calwf3 Output

Using RAW as input:

- `flt.fits`: output calibrated exposure.
- `ima.fits`: output ramp calibration IR exposure.

Using ASN as input with WF3REJ:

- `crj.fits`: cosmic ray rejected image

WF32D

Use this function to facilitate batch runs or for the TEAL interface.

The wf32d primary functions include:

- dark current subtraction
- flat-fielding
- photometric keyword calculations

Only those steps with a switch value of **PERFORM** in the input files will be executed, after which the switch will be set to **COMPLETE** in the corresponding output files.

2.1 Examples

In Python without TEAL:

```
>>> from wfc3tools import wf32d
>>> calwf3.wf32d(filename)
```

In Python with TEAL:

```
>>> from stsci.tools import teal
>>> from wfc3tools import wf32d
>>> teal.teal('wf32d')
```

In Pyraf:

```
>>> import wfc3tools
>>> epar wf32d
```

2.2 Parameters

input

[str] Name of input files

- a single filename (`iaa012wdq_raw.fits`)
- a Python list of filenames
- a partial filename with wildcards (`*raw.fits`)

- filename of an ASN table (`*asn.fits`)
- an at-file (`@input`)

output: str

Name of the output FITS file.

dqicorr: str, “PERFORM/OMIT”, optional

Update the dq array from bad pixel table

darkcorr: str, “PERFORM/OMIT”, optional

Subtract the dark image

flatcorr: str, “PERFORM/OMIT”, optional

Multiply by the flatfield image

shadcorr: str, “PERFORM/OMIT”, optional

Correct for shutter shading (CCD)

photcorr: str, “PERFORM/OMIT”, optional

Update photometry keywords in the header

verbose: bool, optional

Print verbose time stamps?

quiet: bool, optional

Print messages only to trailer file?

The wf32d function can also be called directly from the OS command line:

```
>>> wf32d.e input output [-options]
```

Where the OS options include:

- -v: verbose
- -f: print time stamps
- -d: debug
- -dark: perform dark subtraction
- -dqi: update the DQ array
- -flat: perform flat correction
- -shad: perform shading correction
- -phot: perform phot correction

WF3CCD

This routine contains the initial processing steps for all the WFC3 UVIS channel data. These steps are:

- dqicorr - initializing the data quality array
- atodcorr - perform the a to d conversion correction
- blevcorr - subtract the bias level from the overscan region
- biascorr - subtract the bias image
- flshcorr - subtract the post-flash image

If blevcorr is performed the output contains the overcan-trimmed region.

Only those steps with a switch value of `PERFORM` in the input files will be executed, after which the switch will be set to `COMPLETE` in the corresponding output files.

3.1 Example

In Python without TEAL:

```
>>> from wfc3tools import wf3ccd
>>> calwf3.wf3ccd(filename)
```

In Python with TEAL:

```
>>> from stsci.tools import teal
>>> from wfc3tools import wf3ccd
>>> teal.teal('wf3ccd')
```

In Pyraf:

```
>>> import wfc3tools
>>> epar wf3ccd
```

3.2 Parameters

input

[str] Name of input files

- a single filename (`iaa012wdq_raw.fits`)
- a Python list of filenames

- a partial filename with wildcards (`*raw.fits`)
- filename of an ASN table (`*asn.fits`)
- an at-file (`@input`)

output: str

Name of the output FITS file.

dqicorr: str, “PERFORM/OMIT”, optional

Update the dq array from bad pixel table

atodcorr: str, “PERFORM/OMIT”, optional

Analog to digital correction

blevcorr: str, “PERFORM/OMIT”, optional

Subtract bias from overscan regions

biascorr: str, “PERFORM/OMIT”, optional

Subtract bias image

flashcorr: str, “PERFORM/OMIT”, optional

Subtract post-flash image

verbose: bool, optional

Print verbose time stamps?

quiet: bool, optional

Print messages only to trailer file?

The wf3ccd function can also be called directly from the OS command line:

```
>>> wf32ccd.e input output [-options]
```

Where the OS options include:

- -v: verbose
- -f: print time stamps
- -dq: update the DQ array
- -atod: perform gain correction
- -blev: subtract bias from overscan
- -bias: perform bias correction
- -flash: remove post-flash image

WF3IR

Use this function to facilitate batch runs or for the TEAL interface.

This routine contains all the instrumental calibration steps for WFC3 IR channel images. The steps are:

- dqicorr - initialize the data quality array
- zsigcorr - estimate the amount of signal in the zeroth-read
- blevcorr - subtract the bias level from the reference pixels
- zoffcorr - subtract the zeroth read image
- nlincorr - correct for detector non-linear response
- darkcorr - subtract the dark current image
- photcorr - compute the photometric keyword values
- unitcorr - convert to units of count rate
- crcorr - fit accumulating signal and identify the cr hits
- flatcorr - divide by the flatfield images and apply gain conversion

The output images include the calibrated image ramp (ima file) and the accumulated ramp image (flt file)

Only those steps with a switch value of `PERFORM` in the input files will be executed, after which the switch will be set to `COMPLETE` in the corresponding output files.

4.1 Examples

In Python without TEAL:

```
>>> from wfc3tools import wf3ir
>>> calwf3.wf3ir(filename)
```

In Python with TEAL:

```
>>> from stsci.tools import teal
>>> from wfc3tools import wf3ir
>>> teal.teal('wf3ir')
```

In Pyraf:

```
>>> import wfc3tools
>>> epar wf3ir
```

4.2 Parameters

input

[str] Name of input files

- a single filename (`iaa012wdq_raw.fits`)
- a Python list of filenames
- a partial filename with wildcards (`*raw.fits`)
- filename of an ASN table (`*asn.fits`)
- an at-file (`@input`)

output: str

Name of the output FITS file.

verbose: bool, optional

Print verbose time stamps?

quiet: bool, optional

Print messages only to trailer file?

The wf3ir function can also be called directly from the OS command line:

```
>>> wf32ir.e input output [-options]
```

Where the OS options include:

- `-v`: verbose
- `-f`: print time stamps

WF3REJ

This calls the wf3rej executable. Use this function to facilitate batch runs or for the TEAL interface.

5.1 Example

In Python without TEAL:

```
>>> from wfc3tools import wf3rej
>>> wf3rej.wf3rej(filename)
```

In Python with TEAL:

```
>>> from stsci.tools import teal
>>> from wfc3tools import wf3rej
>>> teal.teal('wf3rej')
```

In Pyraf:

```
>>> import wfc3tools
>>> epar wf3rej
```

5.2 Parameters

input : str, Name of input files

- a single filename (iaa012wdq_raw.fits)
- a Python list of filenames
- a partial filename with wildcards (*raw.fits)
- filename of an ASN table (*asn.fits)
- an at-file (@input)

output : str, Name of the output FITS file.

crrejtab : string, reference file name

scalense : string, scale factor applied to noise

initgues : string, initial value estimate scheme (minlmed)

skysub : string, how to compute the sky (nonelmodelmean)

crsigmas : string, rejection levels in each iteration
crradius : float, cosmic ray expansion radius in pixels
crthresh : float, rejection propagation threshold
badinpdq : int, data quality flag bits to reject
crmask : bool, flag CR in input DQ images?
shadcorr : bool, perform shading shutter correction?
verbose : bool, optional, Print verbose time stamps?

The wf3rej executable can also be called directly from the OS command line prompt:

```
>>> wf3rej.e input output [-options]
```

Input can be a single file, or a comma-delimited list of files.

Where the OS options include:

- t: print the timestamps
- v: verbose
- shadcorr: perform shading shutter correction?
- crmask: flag CR in input DQ images?
- table <filename>: the crreftab filename
- scale <number>: scale factor for noise
- init <medlmin>: initial value estimate scheme
- sky <nonelmedianlmode>: how to compute sky
- sigmas: rejection levels for each iteration
- radius <number>: CR expansion radius
- thresh <number> : rejection propagation threshold
- pdq <number>: data quality flag bits to reject

RUNNING ASTRODRIZZLE

runastrodriz is a module to control operation of astrodrizzle which removes distortion and combines HST images in the pipeline.

6.1 Typical Usage

```
>>> runastrodriz.py [-fhibn] inputFilename [newpath]
```

6.2 Alternative Usage

```
>>> python
>>> from wfc3tools import runastrodriz
>>> runastrodriz.process(inputFilename, force=False, newpath=None, inmemory=False)
```

6.3 GUI Usage under Python

```
>>> python
>>> from stsci.tools import teal
>>> import wfc3tools
>>> cfg = teal.teal('runastrodriz')
```

6.4 PyRAF Usage

```
>>> epar runastrodriz
```

6.5 Options

If the '-i' option gets specified, no intermediate products will be written out to disk. These products, instead, will be kept in memory. This includes all single drizzle products (*single_sci* and *single_wht*), median image, blot images, and crmask images. The use of this option will therefore require significantly more memory than usual to process the data.

If a value has been provided for the newpath parameter, all processing will be performed in that directory/ramdisk. The steps involved are:

- create a temporary directory under that directory named after the input file
- copy all files related to the input to that new directory
- change to that new directory and run astrodrizzle
- change back to original directory
- move (not copy) ALL files from temp directory to original directory
- delete temp sub-directory

The '-b' option will run this task in BASIC mode without creating headerlets for each input image.

The '-n' option allows the user to specify the number of cores to be used in running AstroDrizzle.

SAMPINFO

Sampinfo prints information about a WFC3/IR MultiAccum image, including exposure time information for the individual samples (readouts). The global information listed (and the names of the header keywords from which it is retrieved) includes:

- the total number of image extensions in the file (NEXTEND)
- the name of the MultiAccum exposure sample sequence (SAMP_SEQ)
- the total number of samples, including the “zeroth” read (NSAMP)
- the total exposure time of the observation (EXPTIME).

Information that is listed for each sample is the IMSET number (EXTVER), the sample number (SAMPNUM), the sample time, which is the total accumulated exposure time for a sample (SAMPTIME), and the delta time, which is the additional exposure time accumulated since the previous sample (DELTATIM).

Note that the samples of a MultiAccum exposure are stored in the FITS file in reverse time order. The initial, or “zeroth” read, appears last in the FITS file, with IMSET=NSAMP, SAMPNUM=0, SAMPTIME=0, and DELTATIM=0. The final read of the exposure appears first in the file and has IMSET=1, SAMPNUM=NSAMP-1 (SAMPNUM is zero-indexed), and SAMPTIME=EXPTIME.

7.1 Options

This version will accept a single image name or a python list of images. The list of images should be a python style list, such as:

```
>>> ["image1.fits", "image2.fits"]
```

`add_keys=list()`: You can also supply a supplemental list of keywords to print for each sample, if the key isn’t found in the sample the global header will be checked. If a key is not found the “NA” string will be printed. Additionally you can ask for the median or mean of the data values for each sample using the appropriate switch.

`median=False`: Set to True in order to report the median pixel value for each sample

`mean=False`: Set to True in order to report the mean pixel value for each sample (as measured with `np.min` and `np.max`)

7.2 Usage

Typical:

```
>>> python
>>> from wfc3tools import sampinfo
>>> sampinfo.sampinfo(imagename)
```

Where imagename can be a single filename or a python list() of names

To get the median value for each sample:

```
>>> sampinfo.sampinfo(imagename, median=True)
```

To print additional keys for information:

```
>>> sampinfo.sampinfo(imagename, add_keys=["DETECTOR"])
```

To get the average value for each sample:

```
>>> sampinfo.sampinfo(imagename, mean=True)
```

7.3 Example Output

Default output:

```
In [3]: wfc3tools.sampinfo('ibcf02faq_raw.fits')
IMAGE           NEXTEND SAMP_SEQ           NSAMP  EXPTIME
ibcf02faq_raw.fits  80      STEP50              16      499.234009

IMSET          SAMPNUM SAMPTIME           DELTATIM
1   15          499.234009          50.000412
2   14          449.233582          50.000412
3   13          399.233154          50.000412
4   12          349.232727          50.000412
5   11          299.2323            50.000412
6   10          249.231873          50.000412
7   9           199.231461          50.000412
8   8           149.231049          50.000412
9   7           99.230637           50.000412
10  6           49.230225           25.000511
11  5           24.229715           12.500551
12  4           11.729164           2.932291
13  3           8.796873            2.932291
14  2           5.864582            2.932291
15  1           2.932291            2.932291
16  0           0.0                0.0
```

with median=True:

```
In [4]: wfc3tools.sampinfo('ibcf02faq_raw.fits', median=True)
IMAGE           NEXTEND SAMP_SEQ           NSAMP  EXPTIME
ibcf02faq_raw.fits  80      STEP50              16      499.234009

IMSET          SAMPNUM SAMPTIME           DELTATIM
1   15          499.234009          50.000412          MedPixel: 11384.0
2   14          449.233582          50.000412          MedPixel: 11360.0
3   13          399.233154          50.000412          MedPixel: 11335.0
```

4	12	349.232727	50.000412	MedPixel: 11309.0
5	11	299.2323	50.000412	MedPixel: 11283.0
6	10	249.231873	50.000412	MedPixel: 11256.0
7	9	199.231461	50.000412	MedPixel: 11228.0
8	8	149.231049	50.000412	MedPixel: 11198.0
9	7	99.230637	50.000412	MedPixel: 11166.0
10	6	49.230225	25.000511	MedPixel: 11131.0
11	5	24.229715	12.500551	MedPixel: 11111.0
12	4	11.729164	2.932291	MedPixel: 11099.0
13	3	8.796873	2.932291	MedPixel: 11097.0
14	2	5.864582	2.932291	MedPixel: 11093.0
15	1	2.932291	2.932291	MedPixel: 11090.0
16	0	0.0 0.0	MedPixel: 11087.0	

PSTAT

Plot statistics for a specified image section up the stack of an IR MultiAccum image. Sections from any of the SCI, ERR, DQ, image extensions can be plotted. A choice of mean, median, mode, standard deviation, minimum and maximum statistics is available. The total number of samples is determined from the primary header keyword NSAMP and all samples (excluding the zeroth-read) are plotted. The SCI, ERR, DQ statistics are plotted as a function of sample time. The sample times are read from the SAMPTIME keyword in the SCI header for each readout.

SAMP and TIME aren't generally populated until the FLT image stage To plot the samptime vs sample, use wfc3tools.pstat and the "time" extension

The plotting data is returned as two arrays

8.1 Parameters

filename [file]

Input MultiAccum image name with optional image section specification. If no image section is specified, the entire image is used. This should be either a `_raw` or `_ima` file, containing all the data from multiple readouts. You must specify just the file name and image section, with no extname designation.

extname = "sci" [string, allowed values: sci | err | dq]

Extension name (EXTNAME keyword value) of data to plot.

units = "counts" [string, allowed values: counts | rate]

Plot "sci" or "err" data in units of counts or countrate ("rate"). Input data can be in either unit; conversion will be performed automatically. Ignored when plotting "dq", "samp", or "time" data.

stat = "midpt" [string, allowed values: mean|midpt|mode|stddev|min|max]

Type of statistic to compute.

title = "" [string]

Title for the plot. If left blank, the name of the input image, appended with the extname and image section, is used.

xlabel = "" [string]

Label for the X-axis of the plot. If left blank, a suitable default is generated.

ylabel = "" [string]

Label for the Y-axis of the plot. If left blank, a suitable default based on the plot units and the extname of the data is generated.

plot = True [bool] set plot to false if you only want the data returned

8.2 Usage

pstat.py inputFilename [pixel range]

```
>>> python
```

```
>>> from wfc3tools import pstat
```

```
>>> pstat.pstat(inputFilename,extname="sci",units="counts",stat="midpt",title="",xlabel="",ylabel="")
```

PSTACK

Plot the stack of MultiAccum sample values for a specified pixel in IR multiaccum image. Pixels from any of the SCI, ERR, DQ, or TIME image extensions can be plotted. The total number of samples is determined from the primary header keyword NSAMP and all samples (excluding the zeroth read) are plotted. The SCI, ERR, DQ, values are plotted as a function of sample time, while TIME values are plotted as a function of sample number. The sample times are read from the SAMPTIME keyword in the SCI header for each readout. If any of the ERR, DQ, SAMP, or TIME extensions have null data arrays, the value of the PIXVALUE extension header keyword is substituted for the pixel values. The plotted data values can be saved to an output text table or printed to the terminal.

The BUNIT keyword value is used to determine the starting units of the data.

pstack returns the x and y arrays plotted

9.1 Parameters

input [file]

Input MultiAccum image name. This should be either a `_ima` or `_raw` file, containing all the data from multiple readouts. You must specify just the file name, with no extension designation.

col [integer]

The column index of the pixel to be plotted.

row [integer]

The row index of the pixel to be plotted.

extname = "sci" [string, allowed values: sci | err | dq | samp | time]

Extension name (EXTNAME keyword value) of data to plot.

units = "counts" [string, allowed values: counts | rate]

Plot "sci" or "err" data in units of counts or countrate ("rate"). Input data can be in either unit; conversion will be performed automatically. Ignored when plotting "dq", "samp", or "time" data.

title = "" [string]

Title for the plot. If left blank, the name of the input image, appended with the extname and column and row being plotted, is used.

xlabel = "" [string]

Label for the X-axis of the plot. If left blank, a suitable default is generated.

ylabel = "" [string]

Label for the Y-axis of the plot. If left blank, a suitable default based on the plot units and the extname of the data is generated.

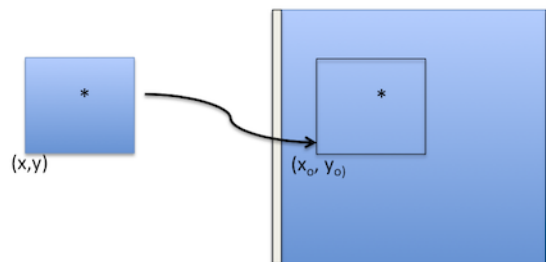
plot = True [bool] set plot to false if you only want the data returned

9.2 Usage

```
>>> python
>>> from wfc3tools import pstack
>>> xdata,ydata=pstack.pstack(inputFilename,column=x,row=y,extname="sci",units="counts|rate",title="
```

SUB2FULL

Given an image specified by the user which contains a subarray readout, return the location of the corner of the subarray in a full frame reference image (including the full physical extent of the chip), in 1-indexed pixels. If the user supplies an X and Y coordinate, then the translated location of that point will be returned. The



10.1 USAGE

```
>>> python
>>> from wfc3tools import sub2full
>>> coords=sub2full.sub2full(filename,x=None, y=None,fullExtent=False)
```

10.2 PARAMETERS

filename [file]

Input image name or list of image names. The rootname will be used to find the _SPT files in the same directory, the SPT file has all the necessary information for the transform.

x = None [integer] Optional

Specify an x coordinate in the subarray to translate

If an x and y are specified, the fullExtent option is turned off and only the translated x,y coords are returned

y = None [integer] Optional

Specify a y coordinate in the subarray to translate If an x and y are specified, the fullExtent option is turned off and only the translated x,y coords are returned

fullExtent = False [bool] Optional

If set, the returned values will include the full extent of the subarray in the reference image

For example: (x0,x1,y0,y1)

10.3 RETURNS

A list of tuples which specify the translated coordinates, either (x0,y0) for each image or the full extent sections

10.4 Example Output

Default output:

```
> sub2full('ibbsolfdqflt.fits')
> [(3584.0, 1539)]
```

Optional output:

Specify a list of images:

```
>im = ['ic5p02e0q_spt.fits',
        'ic5p02e1q_spt.fits',
        'ic5p02e2q_spt.fits',
        'ic5p02e3q_spt.fits',
        'ic5p02e4q_spt.fits']
>
> sub2full(im)
> [(1062.0, 1363),
> (1062.0, 1363),
> (1410.0, 1243),
> (1410.0, 1243),
> (1402.0, 1539)]
```

Return the full extent of the subarray:

```
> sub2full('ibbsolfdqflt.fits',fullExtent=True)
> [(3584.0, 4096, 1539, 2050)]
```

10.5 More information on header keywords

The task uses header keywords from the SPT file of the associated image in order to calculate the offset for the subarray. The keywords it uses are:

Keyword	Meaning
SS_DTCTR	To get the detector for the image
SS_SUBAR	To make sure the image is a subarray
XCORNER	The x corner of the subarray
YCORNER	The y corner of the subarray
NUMROWS	Subarray x size
NUMCOLS	Subarray y size

The UVIS full frame detector has 2051 rows, with 25 pixels of serial overscan. The IR detector has 1024 rows and 5 pixels of overscan.

EMBEDSUB

Given an image specified by the user which contains a subarray readout, return a full-frame image with the subarray implanted at the appropriate location.

11.1 USAGE

```
>>> python
>>> from wfc3tools import embedsub
>>> embedsub.embedsub(files)
```

11.2 PARAMETERS

files [file]

Input image name or list of image names. The rootname will be used to create the output name

11.3 RETURNS

Return the full-frame location of the subarray coordinates using a file specified by the user.

11.4 Example Output

Default output:

```
> wfc3tools.embedsub.embedsub('ic5p02eeq_flt.fits')
> Subarray image section [x1,x2,y1,y2] = [2828:3339,215:726]
> Image saved to: ic5p02eef_flt.fits
```

This method calls `wfc3tools.sub2full` to calculation the subarray position.

SOFTWARE UPDATE HISTORY FOR HSTCAL.CALWF3

Updates for Version 3.1.6 15-Nov-2013 MLS:

- Fixed a file i/o issue after change in cfitsio interaction (see #970, #1073 and #1069)

Updates for Version 3.1.5 30-Sep-2013 MLS:

- Fixed the individual task executables for wf3ir, wf3ccd, wf32d to properly used the user specified output filename when they are called standalone

Updates for Version 3.1.4 09-Sep-2013 MLS:

- Added a couple new functions to deal with user specified subarrays when they start in amp A or C and continue to B or D. In these cases the virtual overscan from the reference postflash file must be avoided, and just incrementing the starting pixel for the array in not a good solution.

Updates for Version 3.1.3 26-Mar-2013 MLS:

- Updated the postflash routine to apply the correct offset for all amps when a user specified subarray is used (no GO users are allowed to do this)
- Some unrelated files will change because I formatted the indentation to make the code easier to decipher

Updates for Version 3.1.2 11-Feb-2013 - MLS:

- Updated the bias subtraction to check for CCDAMP values of SINGLE_AMP and SINGLE_OR_ALL in the reference bias file image when a full frame reference file and a user specified subarray are used so that the correct overscan region is ignored
- Removed check for TDFTRANS per team request, see #980, I'm keeping the same version as the previous change because I havent delivered it yet

Updates for Version 3.1.1 2-Jan-2013 - MLS:

- File I/O in acsrej updated to avoid problems with CFITSIO upcasting file permissions to read/write when not needed. This will allow the hstio code to remove logic that allowed the old code to work but caused problems for CADK when CFITSIO opened ref files in read/write mode because of that hstio logic.

Updates for version 3.1 31-Dec-2012 MLS:

- fixed TrlBufInit problem so it initializes correctly (r21162)

Updates for version 3.1 28-Dec-2012 MLS:

- Updated to account for a memory leak on linux machines during BuildDth when RPTCORR is off and a new spt is being constructed (#967)

<p>Warning: HST CAL DELIVERED, STSDAS+IRAF version no longer maintained, use WFC3TOOLS in HST-CAL</p>
--

Updates 18Oct 2012 - MLS - Version 2.7.1

- fixed a memory leak in cridcalc that was occurring on linux machines and only affected IR data.
- version date and number updated

Updates for version 2.7 21-May-2012 MLS:

- **cridcal.c/wf3dq.h:**
 - update to help negative cr detections (fabs the comparison)
 - updated the spike flag to 1024 so that those pixels weren't ignored in the rejection routine
 - Use zero read pixel value for WF3 IR ramp fitting when saturated
- **do2d.c, cr_scaling.c:**
 - update for BUNIT keyword value so it's not case sensitive, BUNIT value now stored as ELEC-TRONS instead of electrons as well
- wf32d: version update to 07may2012
- wf3rej.cl: version update to 07may2012
- wf3version.h: version update to 07may2012
- wf3main.c: new option r added to print current version and exit

Updates for version 2.6.3 23-Mar-2012 (HAB):

- calwf3.cl: Increment version to 23Mar2012.
- wf3version.h: Increment version to 2.6.3 and date to 23-Mar-2012.
- calwf3/calwf3.c: Upgraded the BuildDthInput function to build file list from names of individual association members when a CRJ sub-product has not been created. (PR 70922; Trac #869)
- calwf3/procir.c: Updated to set CRJ sub-product status to PRESENT after running wf3rej, and report RPTCORR switch status via trlmessage when wf3rej is not run. (PR 70922; Trac #869)

Updates for version 2.6.2 27-Jan-2012 MLS:

- calwf3.cl: Increment version to 27Jan2012.
- wf3version.h: Increment version to 2.6.2 and date to 27-Jan-2012.

- wf3rej/rej.h: Decreased MAX_FILES from 250 to 120 because OPUS is still getting errors when trying to process this many images.

Updates for version 2.6.1 24-Jan-2012 MLS:

- calwf3.cl: Increment version to 24Jan2012.
- wf3version.h: Increment version to 2.6.1 and date to 24-Jan-2012.
- calwf3/procir.c: Added a check for the number of images present when RPTCORR=PERFORM so that wf3rej is not run for singletons.

Updates for version 2.6 - 15-Dec-2011 (HAB):

- calwf3.cl: Increment version to 15Dec2011.
- wf3version.h: Increment version to 2.6 and date to 15-Dec-2011.
- wf3rej/cr_scaling.c: Upgraded to read BUNIT keyword value from first SCI extension header of each input image. (PR 69969; Trac #814)
- wf3rej/rej_do.c: Upgraded to pass new bunit array to and from all functions that need it, in order to handle input data that are in count rates. (PR 69969; Trac #814)
- wf3rej/rej_init.c: Upgraded to rescale input data that are in units of count rates.(PR 69969; Trac #814)
- wf3rej/rej_loop.c: Upgraded to rescale input data that are in units of count rates. (PR 69969; Trac #814)
- wf3rej/rej_sky.c: Upgraded to rescale input data that are in units of count rates. (PR 69969; Trac #814)

Updates for version 2.5.1 - 09-Dec-2011 (HAB):

- calwf3.cl: Increment version to 09Dec2011.
- wf3version.h: Increment version to 2.5.1 and date to 09-Dec-2011.
- calwf3/procir.c: Modified the logic that controls the rptcorr processing so that it's based on the setting of the RPTCORR header keyword switch, instead of just always applying it to every repeat-obs association. (PR 69952; Trac #807)

Updates for version 2.5 - 01-Oct-2011 (HAB):

- calwf3.cl: Increment version to 01Oct2011.
- wf3version.h: Increment version to 2.5 and date to 01-Oct-2011.
- wf3ir/cridcalc.c: Fixed fitsamps routine to correctly accumulate int_time in odd cases where the 1st or 2nd read is bad. (PR 69230; Trac #770)
- wf3ir/rej.h: Increase MAX_FILES from 120 to 250. (PR 63555)
- wf3rej/rej.h: Increased MAX_FILES from 120 to 250. (PR 63555)

Updates for version 2.4.1 - 02-Aug-2011 (HAB):

- calwf3.cl: Increment version to 02Aug2011.
- wf3version.h: Increment version to 2.4.1 and date to 02-Aug-2011.
- lib/wf3info.c: Fixed the logic in the CheckGain routine so that the ref image gets closed before returning when keyval=-1. (PR 68983; Trac #745)

- wf3ir/cridcalc.c: Updated crrej to free memory for tot_ADUs before returning. (PR 68993; Trac #748)

Updates for version 2.4 - 17-Jun-2011 (HAB):

- calwf3.cl: Increment version to 17Jun2011.
- wf3version.h: Increment version to 2.4 and date to 17-Jun-2011.
- calwf3/procccd.c: Modified logic involved in handling error returns from WF3Rej so that WF32d processing still takes place for individual exposures if EXPSCORR=PERFORM. (PR 68593; Trac #722)
- wf3rej/rej_init.c: Added missing call to free(ipts) at end.
- wf3rej/wf3rej.c: Fixed error status return from rej_do so that original status gets passed up for use in caller. (PR 68593; Trac #722)

Updates for version 2.3 - 15-Mar-2011 (HAB):

- calwf3.cl: Increment version to 15Mar2011.
- wf3version.h: Increment version to 2.3 and date to 15-Mar-2011.
- calwf3/calwf3.c: Modified CopyFFile routine to update the FILENAME keyword in created output file. (PR 67225; Trac #646)
- wf3ir/doir.c: No longer load dark ref file for zsigcorr. (PR 67728; Trac #681)
- wf3ir/getirflags.c: Removed zsigcorr checks in checkDark routine, because zsigcorr no longer uses the dark. (PR 67728; Trac #681)
- wf3ir/zsigcorr.c: Modified zsigcorr routine to just subtract the super-zero read image from the science zero read image to estimate zero read signal, rather than scaling the difference between the first and zero reads in the science image. This avoids problems with zero read exposure time in subarray exposures. Also eliminated use of dark image. (PR 67728; Trac #681)

Updates for Version 2.2 - 01-Dec-2010 (HAB):

- calwf3.cl: Increment version to 01Dec2010.
- wf3version.h: Increment version to 2.2 and date to 01-Dec-2010.
- calwf3/calwf3.c: Modified CalWf3Run and BuildDthInput to skip processing for sub-products that have < 2 members present, because no sub-product is produced in this case. (PR 66366; Trac #622)
- calwf3/getrefiles.c: Modified GetIRRef to correctly check all IR switches, so that re-entrant processing works correctly. (PR 66081; Trac #608)
- calwf3/wf3dth.c: Modified InitDthTrl to return with no action if the input member list is empty, to handle missing asn members. (PR 66366; Trac #622)
- calwf3/wf3table.c: Modified GetAsnTable to turn off CRCORR/RPTCORR if there aren't any sub-products with > 1 member. (PR 66366; Trac #622)
- lib/tabpedigree.c: When ttopn has a failure, reset status to zero before returning, so that calling routines have a chance to print error messages before shutting down. (PR 65410; Trac #578)
- lib/trlbuf.c: Modified WriteTrlFile to check for non-null pointer before trying to close trailer file (PR 66366; Trac #622).
- wf3ir/cridcalc.c: Changed crrej to always call EstimateDarkandGlow, regardless of darkcorr setting, because for WFC3 we use a static dark value and therefore don't need access to the darkfile. (PR 66081; Trac #608)

- wf3ir/doir.c: Upgraded crimage header updates to include check of flatcorr status when updating BUNIT values. Also modified noisMsg routine to print noiscorr switch value and have trailer message printed from noiscorr routine itself. Both changes are to support re-entrant processing. (PR 66081; Trac #608)
- wf3ir/getirflags.c: Fixed a reference to dqicorr in checkCRRej that should've been crcorr.
- wf3ir/getirsw.c: Modified GetSw routine to not reset cal switches to OMIT if they have a value other than PERFORM, in order to support re-entrant processing where some switches are COMPLETE. (PR 66081; Trac #608)
- wf3ir/groupinfo.c: Upgraded getDataUnits routine to recognize BUNIT values of ELECTRONS, to support re-entrant processing. (PR 66081; Trac #608)
- wf3ir/irhist.c: Upgraded noisIRHistory routine to first check setting of noiscorr switch before adding history keyword, to support re-entrant processing. (PR 66081; Trac #608)
- wf3ir/noiscalc.c: Modified doNoisIR to print trailer message and noiscorr value, and also give a message saying that noiscorr is skipped if noiscalc returns with an error. Noiscalc was modified to see if the ERR array is already populated before doing the calculation, to support re-entrant processing. (Pr 66081; Trac #608)
- wf3ir/pixcheck.c: Updated the WFC3 IR DQ value assignments. (PR 66080; Trac #607)
- wf3ir/unitcorr.c: Upgraded unitcorr routine to check flatcorr status to decide proper units for BUNIT keyword value update, to support re-entrant processing. (PR 66081; Trac #608)
- wf3ir/zsigcorr.c: Modified to no longer call pixOK function before operating on a pixel. Instead, do the calculation for all pixels. (PR 66080; Trac #607)

Updates for Version 2.1 - 15 May 2010 (HAB):

- calwf3.cl, wf32d.cl, wf3ccd.cl, wf3rej.cl, wf3ir.cl: Increment version to 07May2010.
- wf3version.h: Increment version to 2.1 and date to 07-May-2010.
- calwf3/procccd.c: Modified logic and processing flow so that if CRCORR=PERFORM and EXPSCORR=PERFORM, run wf32d on the individual exposures *after* crcorr is complete, so that CR flags inserted into blv_tmp files by crcorr will show up in final flt images produced by wf32d. (PR 64963; Trac #545)
- wf3rej.cl: Modified to place the user-supplied crsigmas param string in quotes when appending to the command line, so that embedded blanks don't cause problems for the parser. (PR 64941; Trac #544)
- wf3rej/readpar.c: Updated the strtor routine to match the one in the calstis lib, which skips over leading and embedded blanks in the string. (PR 64941; Trac #544)
- wf3ir/darkcorr.c: Updated the darkcorr routine to compute and populate the MEANDARK keyword. (PR 65151; Trac #560)
- wf3ir/doir.c: Swapped the execution order of darkcorr and nlincorr, so now nlincorr goes first. (PR 64854; Trac #536)

Updates for Version 2.0 - 08 Mar 2010 (HAB):

- calwf3.cl, wf32d.cl, wf3ir.cl, wf3rej: Increment version to 08Mar2010.
- wf3version.h: Increment version to 2.0 and date to 08-Mar-2010.
- calwf3/wf3dth.c: Eliminated the creation of dummy drz products. Now done with PyDrizzle. (PR 64261; Trac #495)

- lib/mkspt.c: Modified to allow for the case where there are no input spt files, in which case don't try to create or update the output spt header. (PR 64260; Trac #494)
- wf32d/doflat.c: Modified divFlat to use mean_gain for all images, including grisms. (PR 64259; Trac #493)
- wf3ir/blevcorr.c: Swapping order of zsig and blev such that zsig occurs first requires sending zoff image to blevcorr to be processed. (PR 64262; Trac #496)
- **wf3ir/cridcalc.c:**
 - Added check for pixels already saturated in zeroth read (detected by zsigcorr), in which case outputs set to zero.
 - Switch from using commanded ccdgain to mean_gain.
 - Modified linfit to include readnoise in sample weights and Poisson noise from source in final fit uncertainty.
 - Added SPIKE_THRESH in RejSpikes to use a separate rejection threshold from CR thresh.
 - Updated hardwired dark and readnoise to use SMOV results. Some general cleanup. (PR 64630; Trac #518)
- **wf3ir/doir.c:**
 - **Changed order of processing so that doZsig is called before doBlev. This also requires passing zoff image to**
 - * Compute zero-read sample time (sampzero) here instead of in zsigcorr. (PR 63711; Trac #457)
- wf3ir/flatcorr.c: Modified mult_gain to use mean_gain for all images, including grisms. (PR 64259; Trac #493)
- wf3ir/refdata.c: Fixed initialization of maxcrsplit variable.
- wf3ir/unitcorr.c: No longer need to check status of ZSIGCORR before using sampzero, because sampzero is always computed in doIR. (PR 63711; Trac #457)
- **wf3ir/zsigcorr.c:**
 - **Set ZEROSIG DQ values along with SATPIXEL flags. Set and count pixels as saturated in first read if they'**
 - * Moved computation of sampzero into doIR. (PR 63711; Trac #457)

Updates for Version 1.8.1 - 27 Oct 2009 (HAB):

- calwf3.cl, wf3ir.cl: Increment version to 27Oct2009.
- wf3version.h: Increment version to 1.8.1 and date to 27-Oct-2009.
- wf3ir/cridcalc.c: Fixed the crrej routine to use the logical OR of all input flags for the output DQ flag value for pixels that have all samples rejected, instead of simply flagging them all as SATURATED. (PR 63806; Trac #459)

Updates for Version 1.8 - 26 Oct 2009 (HAB):

- calwf3.cl: Increment version to 26Oct2009.
- wf3version.h: Increment version to 1.8 and date to 26-Oct-2009.
- wf3info.h: Added new “mean_gain” parameter to WF3Info structure. (PR 63788; Trac #458)
- lib/getccdtab.c: Added computation of mean_gain to GetCCDTab. (PR 63788; Trac #458)
- lib/wf3info.c: Added initialization of new mean_gain parameter. (PR 63788; Trac #458)
- wf32d/doflat.c: Upgraded divFlat to use mean_gain when applying gain calibration, except for grism images, which still use the amp-dependent gain values.(PR 63788; Trac #458)
- wf3ir/flatcorr.c: Upgraded mult_gain to use mean_gain when applying gain calibration, except for grism images, which still use the amp-dependent gain values. (PR 63788; Trac #458)

Updates for Version 1.7 - 14 Oct 2009 (HAB):

- calwf3.cl: Increment version to 14Oct2009.
- wf3version.h: Increment version to 1.7 and date to 14-Oct-2009.
- wf3ir/cridcalc.c: Updated the crrej routine to use the badinpdq value from the CRREJTAB to set the DQIGNORE constant, which is used to reject samples, rather than having it hardwired to a value in the code. The hardwired value had been set to just SATPIXEL, which meant that pixels flagged with other values such as DETECTORPROB (4), BADZERO (8), HOTPIX (16), and UNSTABLE (32) were not being blanked out in the output fit image. (PR 63556; Trac ticket #454)
- wf3ir/refdata.c: Updated the crrpar_in routine to report the value of badinpdq, now that it’s being used in cridcalc. (PR 63556; Trac ticket #454)

Updates for Version 1.6 - 17 Aug 2009 (HAB):

- calwf3.cl: Increment version to 17Aug2009.
- wf3version.h: Increment version to 1.6 and date to 17-Aug-2009.
- calwf3/wf3dth.c: Updated to set NEXTEND=3 in header of dummy drz file for IR images. (PR 63286; Trac ticket #436)
- ib/mkspt.c: Updated mkNewSpt to count total number of input spt extensions before updating NEXTEND in output spt file. (PR 63286; Trac ticket #436)
- wf3ir/flatcorr.c: Updated flatcorr routine to set BUNIT to electrons after gain correction has been applied. (PR 63063; Trac ticket #435)
- wf3rej/cr_history.c: Updated to set NEXTEND=3 in header of output crj file for IR images. (PR 63286; Trac ticket #436)

Updates for Version 1.5 - 24 Jun 2009 (HAB):

- calwf3.cl: Increment version to 24Jun2009.
- wf3version.h: Increment version to 1.5 and date to 24-Jun-2009.
- calwf3/procccd.c: Added logic to always use CRCORR=PERFORM internally for both CRJ and RPT associations, instead of using CRCORR for one and RPTCORR for the other.
- wf3rej/rej_check.c: Added logic to getampxy routine to reset ampx/ampy to correct values for IR subarray images. (PR 62948; Trac ticket #424)
- wf3rej/rej_sky.c: Commented out print statement that had inadvertently been left active in a previous delivery that was only intended for debugging use.

Updates for Version 1.4.1 - 27 Apr 2009 (HAB):

- calwf3.cl: Increment version to 27Apr2009.
- wf3version.h: Increment version to 1.4.1 and date to 27-Apr-2009.
- wf3ccd/doblev.c: Added verbose mode print statements to indicate the overscan column limits being used in the overscan calculations. (Trac ticket #405)
- wf3ccd/findover.c: Fixed the logic that was used to compute the biassect values when dealing with a subarray that includes the physical overscan on the amp B/D edge of the image. (Trac ticket #405)

Updates for Version 1.4 - 14 Apr 2009 (HAB):

- calwf3.cl: Increment version to 14Apr2009.
- wf3version.h: Increment version to 1.4 and date to 14-Apr-2009.
- lib/interpinfo.c: Added checks to make sure pixel fractions “q” and “p” come out between 0.0 and 1.0. (Trac ticket #325)
- lib/unbin2d.c: Added checks to make sure pixel fractions “q” and “p” come out between 0.0 and 1.0. (Trac ticket #325)
- lib/unbinline.c: Added checks to make sure argument of sqrt() is positive. (Trac #325)
- wf32d/doflat.c: Fixed bugs that were causing the routine to crash when trying to interpolate a binned LFLTFILE and also added the capability to do a direct division into science image if they are the same size. Added forced return if LFLTFILE is binned, until we upgrade the interpolation routines to work better. (Trac ticket #325)
- wf3ir/refdata.c: Fixed bugs in crrpar_in routine for calls to c_tbegti to read value of IRRAMP column in each row of the crrejt看. (Trac ticket #392)
- wf3ir/resistmean.c: Upgraded computations of mean and standard deviation to use double precision variables. Original single-precision calculations were giving different results on different computer platforms. Also did some general code clean-up. (Trac ticket #391)

Updates for Version 1.3 - 13 Mar 2009 (HAB):

- calwf3.cl: Increment version to 13Mar2009.
- wf3version.h: Increment version to 1.3 and date to 13-Mar-2009.
- wf3info.h: Added “crrej” to WF3Info structure for the CRREJTAB ref table, now that it’s being used within calwf3 in wf3ir/cridcalc step. Previously, it was only accessed from within wf3rej. (Trac ticket #352)
- wf3ccd/dobias.c: Updated to compute correct x-offset values for subarrays in the amp B and D quadrants, which need to take into account the columns of serial virtual overscan that are in the middle of a 4-amp bias reference image. (Trac ticket #378)
- **wf3ir/cridcalc.c:**
 - Added use of CRREJTAB to allow user input of CR rejection threshold instead of having it hardwired in the code.
 - Decreased max_CRs from 6 to 4. Reinstated old loop limits code that excludes reference pixels from ramp fitting. Fixed bug in logic that identifies pixels already saturated in first read.

- Don't set HIGH_CURVATURE flag in output DQ arrays, use UNSTABLE instead, and change messages to say UNSTABLE.
- Also don't set ZEROSIG value in output crimage (fit file) DQ array, because those pixels are still OK (assuming no other flag also set).
- Removed unnecessary call to EstimateDarkandGlow at end of processing.
- Fixed calculation of output SAMP and TIME values.
- Fixed bug in logic that identifies pixels with only 1 good sample.
- Fixed bug in computation of "firstgood" and "lastgood" assignments for pixels with no acceptable samples. (Trac tickets #352, 365, 376, 377, 381)
- wf3ir/getirflags.c: Added new checkCRrej routine to check for the existence and correctness of the CR-REJTAB ref table, for use in CRCORR. (Trac ticket #352)
- wf3ir/refdata.c: Added crrpar_in routine to load parameters from CRREJTAB ref table, for use in CRCORR. (Trac ticket #352)

Updates for Version 1.2a - 20 Feb 2009 (HAB):

- calwf3.cl: Increment version to 20Feb2009.
- wf3version.h: Increment version to 1.2a and date to 20-Feb-2009.
- wf3rej/rej_loop.c: Fixed bug in test to exclude flagged pixels from being tested for CR's so that pixels previously marked as SPILL still get tested to see if they qualify as a CR. (PR 62005)

Updates for Version 1.2 - 29 Jan 2009 (HAB):

- calwf3.cl: Increment version to 29Jan2009.
- wf3version.h: Increment version to 1.2 and date to 29-Jan-2009.
- wf3.h: Added new parameter "type" to RefImage and RefTab structures, which contains the value of the FILETYPE keyword for each reference file. (PR 61608)
- wf3dq.h: New WFC3 UVIS and IR DQ flag assignments. (PR 61741)
- lib/div1d.c: Changed CALIBDEFECT macro to BADFLAT, to coincide with WFC3 DQ assignment changes. (PR 61741)
- lib/getkeys.c: Eliminated use of default values for FILTER and CCDGAIN keywords, which means it will now be an error if they are missing. (PR 61608)
- lib/imgpedigree.c: Upgraded to retrieve FILETYPE along with PEDIGREE/DESCRIP keywords. (PR 61608)
- lib/tabpedigree.c: Upgraded to retrieve FILETYPE along with PEDIGREE/DESCRIP keywords, and to retrieve these keywords from the primary HDU, not the table HDU. (PR 61608)
- lib/trlbuf.c: Fixed bug in CloseTrlBuf causing an IRAF segv, which was due to a call to fclose with a NULL pointer as argument. Removed the call. (PR 61164)
- lib/wf3info.c: Added initialization of new 'type' parameter in InitRefImg and InitRefTab. Also added new CheckImgType, CheckTabType, CheckFilter, CheckDetector, and CheckGain routines. (PR 61608)
- lib/whicherror.c: Added case of an invalid ref file to error message. (PR 61608)
- wf32d/dophot.c: Added computation of PHOTFNU keyword value, to be consistent with IR photcorr process. Also removed some old ACS-specific code that is not used for WFC3. (PR 61138)

- wf32d/get2dflags.c: Upgraded all the checkNNNN routines to verify correct FILETYPE for each reference file, as well as correct selection criteria such as DETECTOR, FILTER, and CCDGAIN. (PR 61608)
- wf32d/photmode.c: Modified construction of photmode string to use separate UVIS1/UVIS2 keywords for the CCD chips, to add the new “cal” keyword for UVIS exposures, and to remove the “DN” keyword for IR exposures because they’re now in units of electrons. (PR 61497)
- wf3ccd/blevdrift.c: Upgraded the cleanDriftFit routine to only use the good values returned by VMedianY in the computation of statistics and rejection of outliers in the array of bias values. Also added checks for potential divide-by-zero conditions. (PR 61698)
- wf3ccd/doblev.c: Upgraded the cleanBiasFit routine to only use the good values returned by FindBlev in the computation of statistics and rejection of outliers in the array of bias values. Also added checks for potential divide-by-zero conditions. (PR 61698)
- wf3ccd/getflags.c: Upgraded all the checkNNNN routines to verify the correct FILETYPE for reference file, as well as correct selection criteria such as DETECTOR, FILTER, and CCDGAIN. (PR 61608)
- wf3ir/cridcalc.c: Reinstated code that had been inadvertently removed from the calnica code ported to calwf3, which propagates CR DQ flags to all samples following a hit. (PR 61425)
- wf3ir/dqicorr.c: Updated to check for missing CCDGAIN and CCDAMP columns in BPIXTAB and default to a match with the science data (same logic as in lib/dodqi.c). (PR 61436)
- wf3ir/flatcorr.c: Fixed bug in mult_gain routine that was doing out of bounds array access for subarray images. (PR 61428)
- wf3ir/getirflags.c: Upgraded all of the checkNNNN routines to verify the correct FILETYPE for each reference file, as well as verifying a match with selection criteria such as DETECTOR and FILTER. (PR 61608)
- wf3ir/refdata.c: Removed FILTER check from getFlatImage because that’s now handled by checkFlat in getIRFlags. (PR 61608)
- wf3ir/zsigcorr.c: Fixed bug in zsigcorr routine to compute correct zeroth read exposure time for subarray images, rather than using SAMPZERO keyword value, which is only correct for full-frame images. (PR 61347)
- wf3rej/rej_loop.c: Fixed problems with DQ flags written to input and output DQ arrays, including not setting any SPILL flags (CR only), not setting CR flags in the output CRJ file for pixels that had at least 1 good input, and not propagating CR flags set for one input file into the remaining files in the input list. (PR 61819)

Updates for Version 1.1 - 10 Oct 2008 (HAB):

- calwf3.cl: Increment version to 10Oct2008.
- wf3version.h: Increment version to 1.1 and date to 10-Oct-2008.
- calwf3/procccd.c: Fixed handling of EXPSCORR=PERFORM so that WF32D gets called for all images, and fixed save_tmp setting so that blv_tmp files get deleted after EXPSCORR processing.
- wf32d/doflat.c: Added ‘applygain’ switch to divFlat to turn on/off the gain correction so that the gain will only be used to correct one ref file and not both, otherwise the gain will be applied twice to the science data.
- **wf3ccd/blevdrift.c:**

– Added new routine cleanDriftFit to reject outliers from parallel overscan array before fitting (as in serial ro

* Added readnoise as an input argument to use in cleanDriftFit.

- wf3ccd/blevfit.c: Modified fit report in BlevFit to indicate that results are for the serial overscan fit.
- wf3ccd/doblev.c: Added readnoise as an input argument to BlevDrift. Modified cleanBiasFit to use different clip values on each pass through data.
- wf3ir/flatcorr.c: Upgraded to convert data to units of electrons by multiplying by the gain after flat field has been applied. Uses new function “mult_gain”.
- wf3rej/rej_sky.c: Added capabilities for “mean” sky calculation, using resistmean.

Updates for Version 1.0 - 11 Sep 2008 (HAB):

- calwf3.cl: Increment version to 11Sep2008.
- wf3version.h: Increment version to 1.0 and date to 11-Sep-2008.
- wf3info.h: Added ncoeff and nerr to NlinData structure.
- wf3sizes.h: Removed this old include file, which isn't used anywhere.
- wf3ir/blevcorr.c: Modified to use statistics from all ref pixels in each readout, rather than working quad-by-quad. Uses new statistics module “resistmean”.
- wf3ir/mkpkg: Added new module “resistmean.c” to library list.
- wf3ir/nlincorr.c: Modified to use 3rd-order coeffs and new ncoeff, nerr members of NlinData struct.
- wf3ir/refdata.c: Modified getNlinData and freeNlinData to use new ncoeff, nerr members of NlinData struct.
- wf3ir/resistmean.c: New statistics module to compute resistant mean.

Updates for Version 0.99 - 28 Aug 2008 (HAB):

- calwf3.cl: Increment version to 28Aug2008.
- wf3version.h: Increment version to 0.99 and date to 28-Aug-2008.
- calwf3/wf3init.c: Changed all occurrences of “_dth” to “_drz”.
- calwf3/wf3table.c: Changed all occurrences of “_dth” to “_drz”.
- lib/detnsegn.c: Removed unnecessary old ACS/HRC code and added WFC3/IR functionality.
- lib/findroot.c: Changed all occurrences of “_dth” to “_drz”.
- lib/getccdtab.c: Modified to only reset ampx for UVIS subarrays, not for IR.
- lib/mkspt.c: Changed all occurrences of “_dth” to “_drz”.
- wf3ir/doir.c: Added call to GetGrp at beginning of processing to load LTV offsets.
- wf3ir/noiscalc.c: Upgraded to use separate gain and readnoise values for each amp quadrant of the images. Includes support for IR subarrays.
- wf3rej/rej_do.c: Set non_zero=nimgs for case where all images have exptime=0, so that they'll still process using exptimes reset to 1.
- wf3rej/rej_init.c: Fixed indexing of SQ(noise.val[0]) to SQ(noise.val[k]) in loop over amps so that appropriate readnoise values get used for each amp.

Updates for Version 0.95 - 21 Jul 2008 (MS):

- calwf3.cl: Increment version to 21Jul2008.
- wf3version.h: Increment version to 0.96 and date to 21-Jul-2008.
- wf3ir/cridcal.c: Major rewrite to incorporate new CR rejection and err computation methods from latest calnica/n_cridcalc.c.

Updates for Version 0.9 - 19 Jun 2008 (HAB):

- calwf3.cl: Increment version to 19Jun2008.
- wf3version.h: Increment version to 0.9 and date to 19-Jun-2008.
- calwf3/procir.c: Added logic and supporting functionality to call WF3Rej_0 to combine IR Repeat-Obs images into a crj product.
- calwf3/wf3dth.c: Restored old acsdth code for creating dummy dth output products, until MultiDrizzle capability is added to WFC3 pipeline.
- lib/mkspt.c: Corrected the calculation of the number of extensions in output spt files for WFC3 (IR files have a pair of extensions for each nsamp).
- wf3rej/cr_history.c: Update RPTCORR, instead of CRCORR, for IR images. Required adding detector as input argument.
- wf3rej/rej_do.c: Pass detector to cr_history.
- wf3rej/rej_sky.c: Avoid arithmetic overflow in binning calculations.
- wf3rej/wf3rej.c: Various updates to properly ID and handle IR images.

Updates for Version 0.8 - 21 Dec 2007 (HAB):

- calwf3.cl: Increment version to 21Dec2007.
- wf3dq.h: Change ZEROSIG DQ value from 4096 to 64, to leave 4096 free for Multidrizzle CR flag.
- wf3version.h: Increment version to 0.8 and date to 21-Dec-2007. lib/dodqi.c: Use new FirstLast routine (provided by P. Hodge) to fix problems with indexing in binned images.
- lib/mkspt.c: Added handling of SNAP1 extensions, in addition to UDL extensions, including appropriate mods to output NEXTEND.
- wf3ir/blevcorr.c: Removed code put in place in previous version to swap quad indexes for images processed before a certain date, because all old images have now been reprocessed to latest orientation. Also updated quad numbering scheme to latest (1 in upperleft and going counter-clockwise from there).
- wf3rej/rej_init.c, rej_loop.c, rej_sky.c: Added calls to hstio getHeader before each call to getShortLine, in order to prevent getShortLine from crashing on null input DQ arrays. In order to handle null arrays, getShortLine needs to access the image header.

Updates for Version 0.7 - 09 May 2007 (HAB):

- calwf3.cl: Increment version to 09May2007.
- wf3info.h: Added “subtype” to WF3Info structure for use with IR subarrays.
- wf3version.h: Increment version to 0.7 and date to 09-May-2007.
- calwf3/getinfo.c: Changed default gain for IR channel from 2.0 to 2.5 in GetIRInfo routine.
- lib/dodqi.c: Modified to allow for wildcard values in BPIXTAB Amp, Gain, and Chip columns (following CALACS change).

- lib/getkeys.c: Updated default gain for IR channel from 2.0 to 2.5. Added 'subtype' to list of IR keywords loaded. Changed default sampzero value to 2.911755 sec.
- wf32d/do2d.c: Modified call to PhotMode to use science extension header, rather than primary header, because that's where phot keywords are.
- **wf32d/photmode.c:**
 - Changed UVIS channel detector keyword to always use "UVIS1".
 - Changed use of "A2Dx" gain keyword to "DN" and eliminated use of it for UVIS images because flatfielding leaves them in units of electrons, not counts.
- wf3ir/darkcorr.c: Eliminated use of RebinRef, because we don't want to extract subarrays from a full-frame dark ref image, we want to instead use a matching subarray dark ref image.
- wf3ir/getirflags.c: Added logic to checkDark to turn off zsigcorr if dark=dummy.
- **wf3ir/imageio.c:**
 - **Enhanced copyGroup to only copy filename if input name is not Null.**
 - * Added new putCalDataSect routine.
- **wf3ir/refdata.c:**
 - Reduced ALLOWDIFF from 0.1 to 0.01 for use with IR subarray exptimes.
 - Added check for SUBTYPE in getDarkInfo.
- wf3ir/wf3ir.c: Added use of new putCalDataSect routine to write out calibrated images that have the ref pixels trimmed off.

Updates for Version 0.61 - 01 Aug 2006 (HAB):

- calwf3.cl: Increment version to 01Aug2006.
- wf3version.h: Increment version to 0.61 and date to 01-Aug-2006.
- wf3ccd/doblev.c: Fixed logic used to select the appropriate readnoise value to pass to the FitToOverscan routine and to convert the readnoise value to units of DN, so that it matches the science data.
- wf3ir/blevcorr.c: Enhanced the blevcorr routine to swap the quad indexes around for raw images generated before and after the date on which OPUS starting transposing the raw IR images.
- wf3ir/flatcorr.c: Switched routine from multiplying by (inverse) flats to dividing by flats.
- wf3ir/math.c: Upgraded adiv and adiv_noref routines to avoid divide by zero errors when computing output err values.

Updates for Version 0.6 - 17 Jul 2006 (HAB):**

- calwf3.cl: Increment version to 17Jul2006.
- wf3version.h: Increment version to 0.6 and date to 17-Jul-2006.
- calwf3/calwf3.c: CalWf3Run routine modified to remove updateAsnStat routine, because only OPUS should update the ASN_STAT keyword in asn tables.

- calwf3/procccd.c: ProcessCCD routine modified to use new “wf3rej_msgtext” string variable to hold (potentially) very long list of input file names for printing. Sometimes too long for regular MsgText string variable.
- calwf3/refexist.c: RefExist routine modified to include check for ref file names that are null (“”), in addition to existing check for “N/A”.
- calwf3/wf3dth.c: InitDthTrl routine modified to fix “trl_in” memory allocation problem for holding long list of trailer file names.
- calwf3/wf3table.c: getAsnTable routine modified to only populate sub-products if at least one input exists for that product.
- lib/dodqi.c: DoDQI routine modified to properly handle binned images, and to adjust flagged pixel coords read from BPIXTAB for presence of serial virtual overscan in WFC3 raw images.
- lib/mkoutname.c: MkOutName routine modified to include calls to “free”, to free local memory before all error returns.
- wf3ccd/blevdrift.c: VMedianY routine modified to fix bug in “if”-statement logic being used to reject flagged pixels from the parallel overscan region. Flawed logic was allowing flagged pixels to remain in computation.
- wf3ccd/findblev.c: FindBlev routine modified to fix bug in “if”-statement logic being used to reject flagged pixels from the serial overscan regions. Flawed logic was allowing flagged pixels to remain in computation.

Updates for Version 0.5 - 08 Nov 2005 (HAB):

- calwf3.cl: Increment version to 08Nov2005.
- wf3version.h: Increment version to 0.5 and date to 08-Nov-2005.
- wf32d/do2d.c: Modified logic in OscnTrimmed routine to make it compatible with WFC3 binned images.
- wf3ir/blevcorr.c: Fixed bug in calculation of j2 loop limit for reference pixel regions for quads 3 and 4.
- wf3ir/nlincorr.c: Fixed bug in calculation of nlin ref image pixel indexes.
- wf3ir/noiscalc.c: Fixed bug in noise computation by adding use of “noise2” variable to temporarily store value of readnoise-squared.
- wf3ir/zsigcorr.c: Fixed bug in calculation of nlin ref image pixel indexes.

Updates for Version 0.4 - 14 Feb 2005 (HAB):

- calwf3.cl: Increment version to 14Feb2005.
- wf3rej.cl: Increment version to 14Feb2005.
- wf3version.h: Increment version to 0.4 and date to 14-Feb-2005.
- wf3ccd/findover.c: Enhanced FindOverscan routine to handle IR images differently than UVIS, selecting oscntab row based on image size (nx,ny) instead of binning.
- wf3ir/blevcorr.c: Enhanced to set reference pixel statistics computation limits based on biassect values in oscntab, rather than image trim values.
- wf3rej/wf3rej.c: Fixed memory reallocation in InitRejTrl that was causing a crash for very large numbers of input images. Made reallocation increment much larger, so that it doesn't get called repeatedly.

Updates for Version 0.3 - 20 Feb 2004 (HAB):

- calwf3.cl: Increment version to 0.3.

- wf3.h: Added ATOD_SATURATE macro definition.
- wf3dq.h: Added ATODSAT dq value of 2048 and changed existing ZEROSIG from 2048 to 4096.
- wf3version.h: Incremented version to 0.3 and date to 20-Feb-2004.
- lib/dodqi.c: Modified to make CCDAMP and CCDGAIN columns optional when looking for matching rows in BPIXTAB. Added handling of new ATODSAT dq flag.
- lib/donoise.c: Fixed use of amp boundaries to take into account WFC3 serial virtual overscan regions.
- lib/getccdtab.c: Changed use of wf3->binaxis to wf3->bin to make it work properly for binned science images.
- lib/getgrp.c: Eliminated the ACS practice of hardwiring wf3->bin to 1 and instead populate it by reading BINAXIS keywords from sci extension header.
- lib/getkeys.c: Eliminated attempt to read BINAXIS keywords from primary header because for WFC3 they're in the sci extension header.
- lib/loadhead.c: Minor code cleanup.
- **wf3ccd/doblev.c:**
 - Implemented limit on sdev to be sqrt(mean) for first pass in CleanBiasFit and use readnoise as value of sdev for second pass.
 - Added readnoise ('rn') as input to cleanBiasFit.
- wf3ccd/doccd.c: Minor comment change.
- wf3rej/rej_loop.c: Commented out unused LoadHdr function declaration. Removed SQ(scale*val) from sumvar computation. Changed AllocBitBuff to work with arbitrary buffer sizes rather than only those evenly divisible by 8.

Updates for Version 0.2 - 28 Oct 2003 (HAB):

- **wf3info.h:**
 - **Changed datatype of 'ccdgain' from int to float.**
 - * Added 'blev(NAMPS)' to WF3Info struct so WF3CCD can remember all blev values for all extensions/amps.
 - Added 'expscorr' to WF3Info struct for use in WF32D.
- wf3version.h: Incremented version to 0.2 and date to 28-Oct-2003.
- wf3wild.h: Added 'FLT_WILDCARD' and 'FLT_IGNORE' macros for use in floating-pt get/put keyword functions.
- calwf3/calwf3.h: Changed datatype of 'scigain' from int to float.
- calwf3/calwf3.c: Removed unique code for RPTCORR processing and made it same as CRCORR for UVIS images.
- calwf3/getinfo.c: Changed datatype of 'scigain' values from int to float.
- calwf3/getreffiles.c: Load 'CRREJTAB' ref table if RPTCORR is turned on (to make it same as CRCORR for UVIS images).
- calwf3/getswitches.c: Changed to handle RPTCORR switch the same as CRCORR for UVIS images.

- **calwf3/procccd.c:**
 - **Changed to handle RPTCORR processing same as CRCORR for UVIS images.**
 - * Added check on status value returned from WF3Rej. If set to 'NO_GOOD_DATA', it will reset 'wf3hdr->sci_basic_2d' to 'SKIPPED' so that no further processing will be performed. It then resets the status value to 'WF3_OK' for continuing normally.
- calwf3/wf3table.c: Changed to handle RPTCORR processing same as CRCORR for UVIS images.
- lib/wf3info.c: Added initialization of new wf3->blev array.
- lib/dodqi.c: Updated to treat commanded gain values as float datatype instead of int.
- lib/donoise.c: Added logic to use Amp C/D bias values from new blev array for UVIS Chip 2 instead of relying on 'AMPY' logic.
- lib/getccdtab.c: Updated to treat commanded gain values as float datatype instead of int.
- lib/getkeys.c: Updated to treat commanded gain values as float datatype instead of int.
- lib/mkspt.c: Updated a couple of printf statements to use trlmessage so that the comments on creating the SPT file also make it to the trailer file.
- lib/sameint.c: Added new 'SameFlt' routine for use with gain keyword values.
- lib/trlbuf.c: Increased 'trldata' buffer size from 'SZ_FNAME' to 'SZ_LINE'.
- lib/key.c: Changed putKeyBool function type from Bool to int.
- wf32d/wf32d.c: Added 'expscorr' switch as command-line argument for wf32d.
- wf32d/do2d.c: Update final state of 'expscorr' switch in output header.
- wf32d/photmode.c: Updated to treat gain values as float datatype instead of int.
- wf3ccd/wf3ccd.c: Populate BIASLEV[abcd] keywords in output header using new 'BiasKeywords' function.
- wf3ccd/blevfit.c: Added BlevResults function to return the values of the slope and intercept computed for the bias fit. Also, the fit reports the values to the user in a trailer message.
- wf3ccd/doatod.c: Updated to treat commanded gain values as float datatype instead of int.
- **wf3ccd/doblev.c:**
 - Added 'cleanBiasFit' routine to do sigma-clipping on bias measurements before computing fit.
 - Set default cdbias value to be AMP C/D value for UVIS Chip 2 data where no overscan was available for computing the bias level.
 - Modified to load the 'biasect' array with indexes corresponding to the serial physical overscan regions, instead of serial virtual overscan regions, when processing UVIS subarray images (which have noserial virtual overscan).
- **wf3ccd/doccd.c:**
 - **Added processing msg's giving info on bias levels for each amp.**

- * Upgraded to do correct overscan trimming of output image for UVIS subarray modes, in which there's no serial virtual overscan to remove, and variable amounts of serial physical overscan.
- wf3ccd/findover.c: Modified to zero-out all serial and parallel virtual biassect and trim values when processing UVIS subarray images (which don't have any virtual overscan). Also fixed a bug in which one of the biassect values was not being converted from 1-indexed to 0-indexed in the case of subarray images.
- wf3ir/dqicorr.c: Updated to treat commanded gain values as float datatype instead of int.
- wf3ir/getirflags.c: Modified to load DARKCORR and NLINCORR switch settings and DARKFILE and NLINFILE ref file info if ZSIGCORR is set to PERFORM.
- wf3ir/nlincorr.c: Modified to use just 1 node array from the NLINFILE ref data, which is the saturation value. There won't be another node array specifying the lower bound of the nlin correction as with NICMOS.
- **wf3ir/refdata.c:**
 - Modified to load just 1 node array from the NLINFILE ref file.
 - Also modified to combine all of the PFLT, DFLT, and LFLT ref file data (if present) into a master flat, as is done for UVIS processing.
- wf3ir/zsigcorr.c: Modified to use just 1 node array from the NLINFILE ref data, which is the saturation value.
- wf3rej/wf3rej.c: Added call to 'mkNewSpt' within error condition for wf3rej_do to always produce a new SPT file for product when possible. This also involved remembering the value of the error condition, setting it to WF3_OK, calling 'mkNewSpt', then resetting to old value in order to allow 'mkNewSpt' to work successfully.
- **wf3rej/rej_do.c:**
 - Added code to count number of inputs with exptime>0. If some are zero, new code will insure that first good image gets used to initialize the initial guess image.
 - Revised to handle cases where 0,1,or more input are valid. If none have exptime>0, skips wf3rej_loop altogether and output a blank image with DQ values of 1 and ERR values of 0 with the exception of the 0,0 pixel, which have values of 8 and -1 respectively, to forces HSTIO to write out the image arrays. It now returns status=NO_GOOD_DATA if there are no inputs with exptime>0.
- **wf3rej/rej_init.c:**
 - Added code to count number of inputs with exptime>0.
 - Also now checks whether exptime!=0 when building initial guess image.
- wf3rej/rej_loop.c: Added code to avoid crashing when exp[n]=0 for an input image. It will now skip all the detection code when exp[n]=0.
- wf3rej/cr_scaling.c: Added trailer file comments to better describe how exptime=0 cases are handled.

Updates for Version 0.1 - 26 Nov 2002 (HAB):

- Initial installation of baseline CALWF3 into stlocal\$testwf3 pkg.

INDICES AND TABLES

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