



Technical Instrument Report WFPC2 2008-003

An Update on Creating a WFPC2 Yearly Superbias

Deepashri Thatte, John Biretta
July 29, 2008

ABSTRACT

The procedure for producing a yearly superbias file from one hundred and twenty of the individual bias frames is discussed in this TIR . The individual images are taken during the year as part of the WFPC2 calibration plan. The file is used to calibrate all WFPC2 observations during the following year.

Introduction

A superbias should be produced yearly, using a selection of the bias images from the previous twelve months. The superbias is a cosmic ray rejected and averaged combination of one hundred and twenty biases chosen to be free of severe bias jumps and visual anomalies. It is used in the pipeline calibration of WFPC2 observations taken throughout the year.

I. CREATING THE SUPERBIAS IMAGE

The procedure described below should be carried out separately once for gain 7 and once for gain 15 superbias.

(1) Select and Retrieve Data

Select and retrieve all the gain 7 bias frames with WF4 biaseven > 240 between February 21, 2006 and May 14, 2007.

Start Time = 02/21/2006..05/14/2007
Gain = 7
Serials = OFF
Imagetype = Bias
WF4 biaseven > 240.

In Starview custom query generator right click on wp2_biaseven4 (DATATBASE STRUCTURE > catalog > wfpc2_primary_data > wp2_biaseven_4) and add it to the general search form which can then be used for choosing WF4 biaseven > 240. Use the show override (Expert Only) option in Starview to select just the extensions c0f and c1f.

Make the following changes for gain 15 bias frames:

Gain = 15
WF4 biaseven > 290

Make two different archive requests and retrieve the data in two different directories, one for gain 7 and one for gain 15.

Select Uncalibrated Data for delivery. Once you have the data you can delete all shf.fits and trl.fits images.

```
rm *shf.fits *trl.fits
```

In IRAF

```
set imtype=hhh  
strfits *dof.fits "" ""
```

```
mkdir fits
mv *.fits fits
```

(2) Recalibrate the Raw Files with calwp2

Start IRAFX

```
set uref= /grp/hst/cdbs/uref/ (This is default uref) for Solaris or
set uref= /grp/hst/cdbs/uref_linux/ for Linux
set ucal= /working/directory/where/d0h/files/are/
ucal should be the same as the working directory where the d0h files are.
```

The directory should have q0h, x0h and q1h files in addition to d0h

Make sure that uref directory has all the reference files and ucal directory has all the calibration files shown in the .d0h image header. Edit the image headers to set mask, atod and blevcorr to perform.

The parameters for hedit should look like the following

lpar hedit

```
images = "*.d0h"      images to be edited
fields = "maskcorr,atodcorr,blevcorr" fields to be edited
value = "perform"    value expression
(add = no)          add rather than edit fields
(addonly = no)      add only if field does not exist
(delete = no)       delete rather than edit fields
(verify = yes)      verify each edit operation
(show = yes)        print record of each edit operation
(update = yes)      enable updating of the image header
(mode = "al")
```

Calwp2 recalibrates the bias files. Create a file called cal.cl that has lines like the following:

```
calwp2 u4312801r u4312801r
calwp2 u4312802r u4312802r
calwp2 u4312803r u4312803r
```

stsdas

hst_calib

wfpc

```
ls *d0h >> list.txt
```

```
!nawk '{sub(/.d0h/, " "); if ($0 !~/xxx/) print}' list.txt >> roots.txt
```

```
!awk '{print "calwp2 "$1" "$1}' roots.txt >> cal.cl
```

```
cl < cal.cl
```

(3) Correct for WF4 Anomaly

Process WF4 images with special software to correct the CCD amplifier gains and to remove horizontal (x-direction) streaks from the images. See Appendix A and ISR WFPC2 2005-02.

(4) Check for Bias Jumps

After recalibration, find out which bias images had bias jumps. The jumps are found by *calwp2*, and documented in the history section of the calibrated header file (.c0h). You will be eliminating the images with the largest jumps.

```
!grep level *c0h > temp1.txt
```

```
!grep HISTORY temp1.txt > temp2.txt
```

Reformat the files

```
!nawk '{sub(/WARNING:\/, " ");if($0!~/xxx/)print}' temp2.txt>>temp3.txt
```

(Look for 'WARNING:' in temp2.txt. Replace it by space and write the corrected line in temp3.txt.)

```
!nawk '{sub(/h:HISTORY/, "h");if($0!~/xxx/)print}' temp3.txt >>bjumps.txt
```

(Look for 'h:HISTORY' in temp3.txt. Replace it by h and write the corrected line in bjumps.txt.)

```
!rm temp[1-3].txt
```

bjumps.txt may have images that are listed more than once (because those images had a bias jump in more than one chip). Remove the lines (chips) with the lower bias jump for any repeated image. So we will have a list of all the biases with the largest bias jump in each image being noted. John's script to correct for WF4 streaks makes all the bias jumps in WF4 zero. Therefore all WF4 bias jumps should be treated as '0'.

The following steps will accomplish this.

```
!grep PC1: bjumps.txt >> bjumps1.txt  
!grep WF2: bjumps.txt >> bjumps2.txt  
!grep WF3: bjumps.txt >> bjumps3.txt  
!grep WF4: bjumps.txt >> bjumps4.txt
```

```
tjoin bjumps1.txt bjumps2.txt match1.txt c1 c1 extraro=neither  
tjoin bjumps1.txt bjumps3.txt match2.txt c1 c1 extraro=neither  
tjoin bjumps1.txt bjumps4.txt match3.txt c1 c1 extraro=neither  
tjoin bjumps2.txt bjumps3.txt match4.txt c1 c1 extraro=neither  
tjoin bjumps2.txt bjumps4.txt match5.txt c1 c1 extraro=neither  
tjoin bjumps3.txt bjumps4.txt match6.txt c1 c1 extraro=neither
```

```
ls match?.txt >> matchfiles.txt
```

```
!grep bias match?.txt >> doubles_temp.txt
```

```
tselect doubles_temp.txt doubles1.txt "c6 .gt. c12"  
tselect doubles_temp.txt doubles2.txt "c6 .le. c12"  
tproject doubles1.txt doubles3.txt c1,c8,c12  
tproject doubles2.txt doubles4.txt c1,c2,c6  
cat doubles[3-4].txt >> doubles5.txt
```

doubles.txt has the names of lines you want to remove from *bjumps.txt*

Anything with more than two entries on the bjumps list will be repeated in the doubles list, so be sure to check for more than two copies. For these cases you will need to delete more than one of the lines, leaving only the line with the largest bjump value.

```
!nawk '{sub(/match/, "");if($0!~/xxx/)print}' doubles5.txt >> doubles6.txt  
!nawk '{sub(/1.txt:/, "");if($0!~/xxx/)print}' doubles6.txt >> doubles7.txt  
!nawk '{sub(/2.txt:/, "");if($0!~/xxx/)print}' doubles7.txt >> doubles8.txt  
!nawk '{sub(/3.txt:/, "");if($0!~/xxx/)print}' doubles8.txt >> doubles9.txt  
!nawk '{sub(/4.txt:/, "");if($0!~/xxx/)print}' doubles9.txt >> doubles10.txt  
!nawk '{sub(/5.txt:/, "");if($0!~/xxx/)print}' doubles10.txt >> doubles11.txt  
!nawk '{sub(/6.txt:/, "");if($0!~/xxx/)print}' doubles11.txt >> doubles.txt
```

```
tsort doubles.txt c1
```

Edit *bjumps.txt* (by hand, or by other method)

Remove the lines (chips) with the lower bias jump for any repeated image from *bjumps.txt*. Treat all WF4 bias jumps as '0' since they are corrected.

```
tsort bjumps.txt c1 #remove extra spaces in this file before doing tsort
```

```
!rm bjumps[1-4].txt doubles[1-9].txt doubles1[0-1].txt match[1-6].txt
```

```
!rm matchfiles.txt doubles_temp.txt
```

Check that you have gotten all the doubles or triples in *bjumps.txt*. One check that can be run is to verify that where *doublescheck.txt* has repeated rootnames all the lines have the same rootname. If this condition is not met, you know there is a line that has not yet been deleted in *bjumps.txt* which should be.

```
!awk '{print "\!grep \"$1\" bjumps.txt >> doublescheck.txt"}' < doubles.txt >> grep.cl
```

This gets all the filenames from *doubles.txt* and generates a script to look for these filenames in *bjumps.txt*

```
cl < grep.cl
```

Add images with no recorded bias jumps to the list of usable images (they were not on the original *bjumps* list because *calwp2* had no reason to add a line to those images' headers).

```
ls *c0h > list.txt
```

```
tjoin list.txt bjumps.txt new_bjumps.txt c1 c1 extraro=first
```

Check that there are the same number of lines in *list.txt* as in *new_bjumps.txt*

```
!wc list.txt new_bjumps.txt
```

Sort the list according to bias jump value with the lowest bias jumps at top of list.

```
!nawk '{gsub(/"/, "0");if($0!~/xxx/) print}' new_bjumps.txt >> bjumps_sorted_all.txt
```

Verify that the bias jump levels are in the sixth column (c6) *bjumps_sorted_all.txt* before proceeding.

```
tsort bjumps_sorted_all.txt c6
```

```
tproject bjumps_sorted_all.txt bjumps_sorted.txt c1
```

The list is now sorted in order of severity of bias jumps. You will use only the 120 biases with the lowest bias jumps, minus any disqualified in the visual review. Generally, it's OK to use images with bias jumps less than about 1 DN. If you do not have 120 images with bias jump less than 1 DN, which is unexpected, then you may have to look into other options.

(5) Visually Review Images

Visually examine all groups in all images and write the names of unwanted images in anomalies.txt.

Use the display method in Appendix B to visually examine all the images.

You can expect to see some or all of the following:

a) Horizontal lines, bands, jumps of varying width and location:

These will mostly be caught with the bias jump list you've already compiled. As long as the difference is low level it is not a concern.

b) Some images with a much higher than normal number of cosmic rays and/or uncommonly severe CRs:

You can eliminate these images.

c) A light leak / partial illumination anomaly:

This was seen in biases before 2003 (S. Gonzaga). This problem should be noted and images that are affected should not be used.

d) Persistence (an area of higher signal that may appear for a time after imaging a bright object)

You can eliminate these images if there is a significant difference in levels.

(6) Choose Best 120 Bias Images

Once you have identified large bias jumps and visual anomalies you can choose the 120 best biases to use in the superbias. Make a list of the images you plan to use (with .c0h extensions).

```
copy bjumps_sorted.txt filesused.txt
```

Edit filesused.txt to remove names of unwanted images. Check that filesused.txt has 120 lines.

```
!wc filesused.txt
```

(7) Create Plots to Check for Anomalies

For each chip, create plots for "biaseven vs date" and "biasodd vs. date" to check for anomalies in the global bias level, and make sure the scatter looks reasonable. You can do this for both the full year's worth of biases and for just the 120 that you plan to use. Repeat this for each gain setting (7,15).

In IRAF:

```
hselect *c0h[1] "rootname,expstart,biasodd,biaseven" yes > bias1.txt
hselect *c0h[2] "rootname,expstart,biasodd,biaseven" yes > bias2.txt
hselect *c0h[3] "rootname,expstart,biasodd,biaseven" yes > bias3.txt
hselect *c0h[4] "rootname,expstart,biasodd,biaseven" yes > bias4.txt
!awk '{print "hselect "$1"[1] rootname,expstart,biasodd,biaseven yes >>
bias1used_7.txt"}' filesused.txt >> hselect.cl
!awk '{print "hselect "$1"[2] rootname,expstart,biasodd,biaseven yes >>
bias2used_7.txt"}' filesused.txt >> hselect.cl
!awk '{print "hselect "$1"[3] rootname,expstart,biasodd,biaseven yes >>
bias3used_7.txt"}' filesused.txt >> hselect.cl
!awk '{print "hselect "$1"[4] rootname,expstart,biasodd,biaseven yes >>
bias4used_7.txt"}' filesused.txt >> hselect.cl
cl < hselect.cl
```

A gnuplot script to plot 'biaseven vs date' and 'biasodd vs date' is in Appendix C. You can also use a different plotting tool of your choice. A Supermongo script is in TIR WFPC2 2005-003.

Note that the plots for WF4 do not show corrected biaseven and biasodd values. These values are read from the headers and the headers are not updated after correction for WF4 anomaly. So the plots for WF4 show full effect of WF4 anomaly and are therefore not useful.

(8) Combine Images in Sets of Forty

Combine each group of forty biases using the IRAF mkdark task, creating three new bias files and three new dqf files.

For Gain=7

```
tselect filesused.txt list1_7 "row() .gt. 0 .and. row() .le. 40"
tselect filesused.txt list2_7 "row() .gt. 40 .and. row() .le. 80"
tselect filesused.txt list3_7 "row() .gt. 80 .and. row() .le. 120"
```

```
noisepar.readnoi=0.71
noisepar.gain=7
```

```
mkdark @list1_7 bias7_1.c0h bias7_1.c1h "6,5,5,4" 1 0.5 4096.0 -99 "med"
```

```
mkdark @list2_7 bias7_2.c0h bias7_2.c1h "6,5,5,4" 1 0.5 4096.0 -99 "med"  
mkdark @list3_7 bias7_3.c0h bias7_3.c1h "6,5,5,4" 1 0.5 4096.0 -99 "med"
```

For Gain 15

```
tselect filesused.txt list1_15 "row() .gt. 0 .and. row() .le. 40"  
tselect filesused.txt list2_15 "row() .gt. 40 .and. row() .le. 80"  
tselect filesused.txt list3_15 "row() .gt. 80 .and. row() .le. 120"
```

```
noisepar.readnoi=0.5  
noisepar.gain=15
```

```
mkdark @list1_15 bias15_1.c0h bias15_1.c1h "6,5,5,4" 1 0.5 4096.0 -99 "med"  
mkdark @list2_15 bias15_2.c0h bias15_2.c1h "6,5,5,4" 1 0.5 4096.0 -99 "med"  
mkdark @list3_15 bias15_3.c0h bias15_3.c1h "6,5,5,4" 1 0.5 4096.0 -99 "med"
```

(9) Combine the Three Sets of Forty

Combine the three files created in the previous mkdark steps into the superbias, incorporating all 120 bias files.

For Gain=7

```
imcalc bias7_1.c0h,bias7_2.c0h,bias7_3.c0h bias7.c0h "(im1+im2+im3)/3"
```

For Gain=15

```
imcalc bias15_1.c0h,bias15_2.c0h,bias15_3.c0h bias15.c0h "(im1+im2+im3)/3"
```

(10) Generate Data Quality File

Use the .c1h files created when mkdark was run on the groups of forty biases (Section 8) to make data quality file for the superbias.

The .c1h files created as a by-product of mkdark indicate the following: Pixel values indicate the number of individual bias files used to create the corresponding pixel in the combined bias image. If the value of a pixel in the mask file is thirty, it means that the pixel values from thirty images were used to determine the final pixel value in the combined image (and the pixel was rejected in the remaining ten images).

Next, add the three mask files.

For Gain=7

```
imcalc bias7_1.c1h,bias7_2.c1h,bias7_3.c1h bias7_123.c1h "im1+im2+im3"
```

For Gain=15

```
imcalc bias15_1.c1h,bias15_2.c1h,bias15_3.c1h bias15_123.c1h "im1+im2+im3"
```

(11) Adjust the Data Quality File

To create a data quality file for the superbias, determine a cut-off for the number of pixels from 120 bias files that must have been combined for a given pixel for it to be considered good in the superbias. A superbias pixel created from one hundred or more individual biases is flagged as good (value=0). All other pixels are set to bad (value=2).

For Gain=7

```
imcalc bias7_123.c1h bias7.c1h "if im1 .lt. 100. then 2. else 0."
```

For Gain=15

```
imcalc bias15_123.c1h bias15.c1h "if im1 .lt. 100. then 2. else 0."
```

II. FORMATTING FILES

Before continuing, compare the newly-created superbias and associated data quality files with previous files and verify that the superbias looks as expected.

(1) Generate the Header File

The strategy will be to copy the header from the previous superbias and merely change the parameters related to the new superbias image we just created. Specifically, we will combine the header of the previous year's superbias files (c0h and c1h) with the pixel file of the newly-created superbias or supermask file, respectively.

The previous year's superbias you should use are the fully processed and ready to deliver superbias with endings .b2h and b2d for the DQF and .r2h and .r2d for the pixel data file.

Copy all the files related to the old superbias.

For Gain 7

```
cd /grp/hst/cdbs/uref/  
gcopy p1p1510lu.?2h /user/thatte/wfpc2/superbias/superbias04/gain7/
```

Also copy the old superbias to the working directory

```
gcopy p1p1510lu.?2h /user/thatte/wfpc2/superbias/gain7/
```

```
cd /user/thatte/wfpc2/superbias/gain7/
moveheader p1p1510lu.r2h bias7.c0h superbias7.c0h
moveheader p1p1510lu.b2h bias7.c1h superbias7.c1h
```

For gain15

```
cd /grp/hst/cdbs/uref/
gcopy p1p1510ku.?2h /user/thatte/wfpc2/superbias/superbias04/gain15/
```

Also copy the old superbias to the working directory

```
gcopy p1p1510ku.?2h /user/thatte/wfpc2/superbias/gain15/
```

```
cd /user/thatte/wfpc2/superbias/gain15/
moveheader p1p1510ku.r2h bias15.c0h superbias15.c0h
moveheader p1p1510ku.b2h bias15.c1h superbias15.c1h
```

(2) Edit the Header File

The parameters in the superbias header file and maskfile need to be edited. C. Ritchie wrote a task called dohedit2.cl to do this (see Appendix D).

Copy dohedit2.cl into your working directory

```
cd [working directory]
```

```
# define the task:
task dohedit2 = dohedit2.cl
```

```
# to start task, type:
dohedit2
```

Run the task on the superbias file (.c0h) and the supermask file (.c1h)

Enter the following as prompted:

Input images: [name of superbias file, or superbias mask file]

Input image type (MASK, ATOD, BIAS, SHAD, DARK, FLAT): BIAS

Input atodgain: [7 or 15]

Input shutter (A, B, or ‘ ’): ‘ ‘

Input date (dd/mm/yy): [date file was created]

```
# Gain=7
dohedit2
Input images (*.r?h): superbias7.c0h
Input image type (MASK, ATOD, BIAS, SHAD, DARK, FLAT): BIAS
Input atodgain: 7
Input shutter (A, B, or ‘ ’): ‘ ‘
Input date (dd/mm/yy): ??/??/??
```

dohedit2

Input images (*.r?h): superbias7.c1h

Input image type (MASK, ATOD, BIAS, SHAD, DARK, FLAT): BIAS

Input atodgain (7.): 7k

task dohedit2=dohedit2.cl

Gain=15

dohedit2

Input images (*.r?h): superbias15.c0h

Input image type (MASK, ATOD, BIAS, SHAD, DARK, FLAT): BIAS

Input atodgain: 15

Input shutter (A, B, or ‘ ‘) (): ‘ ‘

Input date (dd/mm/yy): ??/??/??

dohedit2

Input images (*.r?h): superbias15.c1h

Input image type (MASK, ATOD, BIAS, SHAD, DARK, FLAT): BIAS

Input atodgain : 15

Input shutter (A, B, or ‘ ‘) (): ‘ ‘

Input date (dd/mm/yy): ??/??/??

(3) Edit Headers

Each pipeline bias file has a history appended to the header, as well as appropriate pedigree, useafter, and descrip keywords. Edit the headers of your DQF and pixel files:

emacs superbias7.c0h &

emacs superbias7.c1h &

Delete all lines beginning with any of the following:

PEDIGREE=

USEAFTER=

DESCRIP =

HISTORY

Create a new HISTORY to be appended. A history template is shown below. You will edit dates, last year's superbias rootname, the list of files used, and your name. The task stfhistory is used to add the history to the file header.

Make a compact list of all 120 files with filename:- for_hist.txt

Edit the list to appropriate width as shown below.

Create a file called h_comment.txt to contain the updates HISTORY that you will use.

This superbias file was created from the following 120 on-orbit bias frames taken between February 2006 and May 2007. It is not significantly different than superbias plp1510ku.r2h

```
u9jc3108m  u9ly100am  u9jc3107m  u9jc2509m  u9ly0609m
u9ly1008m  u9ly0709m  u9ly1007m  u9wj0909m  u9ly0809m
u9jc2009m  u9wj1108m  u9wj120am  u9wj2708m  u9wj1207m
u9wj110am  u9wj1208m  u9jc1908m  u9wj1107m  u9jc1907m
u9wj090am  u9wj1209m  u9ly0808m  u9ly1108m  u9ur1508m
u9ur1607m  u9wj0707m  u9ur150am  u9wj0908m  u9wj170am
u9ur160am  u9ur1707m  u9wj1709m  u9wj070am  u9ur1509m
u9ur130am  u9ly1809m  u9wj1708m  u9wj1507m  u9ly1507m
u9ly1707m  u9ly2307m  u9ur1307m  u9unbf04m  u9ly230am
u9ur1309m  u9wj1607m  u9wj1109m  u9wj270am  u9wj2707m
u9wj1609m  u9wj2607m  u9wj2508m  u9wj250am  u9wj0709m
u9wj0708m  u9ly1508m  u9ly120am  u9ly1109m  u9ly150am
u9ly2308m  u9ly170am  u9ly1708m  u9ly1009m  u9jc3009m
u9jc3008m  u9jc3007m  u9jc300am  u9jc3109m  u9ly080am
u9unb203m  u9ly060am  u9jc310am  u9ly0607m  u9ly1509m
u9ur1609m  u9ur1807m  u9unb204m  u9ur1808m  u9ur1809m
u9ur180am  u9wj160am  u9ly2309m  u9ur1507m  u9ur1709m
u9wj2509m  u9jc2007m  u9jc1909m  u9ur170am  u9ly0707m
u9ly070am  u9wj2507m  u9ly1207m  u9ly1209m  u9wj1707m
u9jc2507m  u9ly1107m  u9wj2608m  u9ly110am  u9jacf03m
u9ly0608m  u9jacf04m  u9ly1208m  u9jc200am  u9wj1509m
u9ur1708m  u9wj2709m  u9wj1508m  u9wj240am  u9jc250am
u9ur1308m  u9wj1608m  u9jc190am  u9jc2008m  u9ly180am
u9jc3407m  u9jc3408m  u9ur1907m  u9ur190am  u9jc2907m
```

The raw images were retrieved from the archive. All 4 CCDs were recalibrated using CALWP2 Version 2.2, with mask, atod, and blevcorr set to PERFORM. The WF4 images were also processed through special software to correct the CCD amplifier gains and remove horizontal (x-direction) streaks from the images (see ISR WFPC2 2005-02). Only images with WF4 BIASEVEN > 240 were used, so as to avoid large gain corrections.

The STSDAS wfpc "mkdark" task (identical to a version of "crrej" previously used to make reference bias files) was used to combine the calibrated frames and remove cosmic rays. The task was run with four iterations, using sigmas set to 6,5,5,and 4; Pixels are rejected if they are N sigma above or below the initial guess image. The first initial guess image was taken to be the median of the stack; subsequent iterations use the computed average image as the initial guess. In addition, pixels within 1 pixel of a rejected pixel (in a '+' pattern) were discarded if they deviated by more than 2,2,1.5, and 1 sigmas, respectively for each iteration. Pixels using less than 100 input images were marked in the DQF.

On-orbit file was generated and formatted for CDDBS installation on June 28, 2007 at STScI.

Superbias for observations taken after February 21, 2006.

Create a similar file for gain 15 superbias.

(a) Append the new HISTORY to your superbias headers.

Use h_comment.txt from respective directories for gain7 and gain15 superbias.

```
# Gain=7
stfhistory superbias7.c0h @h_comment.txt
stfhistory superbias7.c1h @h_comment.txt
# Gain=15
stfhistory superbias15.c0h @h_comment.txt
stfhistory superbias15.c1h @h_comment.txt
```

(b) Add PEDIGREE keyword values (update dates to first and last date of the biases used in creating the superbias).

```
hedit superbias7.c0h PEDIGREE 'INFLIGHT 21/02/06 - 14/05/2007' add+ verify-
hedit superbias7.c1h PEDIGREE 'INFLIGHT 21/02/06 - 14/05/2007' add+ verify-
hedit superbias15.c0h PEDIGREE 'INFLIGHT 21/02/06 -07/05/2007' add+ verify-
hedit superbias15.c1h PEDIGREE 'INFLIGHT 21/02/06 -07/05/2007' add+ verify-
```

(c) Add the USEAFTER keyword, based on the start date intended for the superbias.

```
hedit superbias7.c0h useafter 'February 21, 2006 00:00:00' add+ verify-
hedit superbias7.c1h useafter 'February 21, 2006 00:00:00' add+ verify-
hedit superbias15.c0h useafter 'February 21, 2006 00:00:00' add+ verify-
hedit superbias15.c1h useafter 'February 21, 2006 00:00:00' add+ verify-
```

(d) Add a keyword called DESCRIP. Use the name of the previous year's superbias rootname (where is says not significantly different from....) and the correct dates:

```
hedit superbias7.c0h DESCRIP 'Feb 2006-May 2007 yearly superbias' add+ verify-
hedit superbias7.c1h DESCRIP 'Feb 2006-May 2007 yearly superbias' add+ verify-
hedit superbias15.c0h DESCRIP 'Feb 2006-May 2007 yearly superbias' add+ verify-
hedit superbias15.c1h DESCRIP 'Feb 2006-May 2007 yearly superbias' add+ verify-
```

(e) Add a keyword called COMMENT, using your own name.

```
hedit superbias7.c0h comment 'WFPC2 Yearly Superbias created by D. Thatte' add+ verify-
hedit superbias7.c1h comment 'WFPC2 Yearly Superbias created by D. Thatte' add+ verify-
hedit superbias15.c0h comment 'WFPC2 Yearly Superbias created by D. Thatte' add+ verify-
hedit superbias15.c1h comment 'WFPC2 Yearly Superbias created by D. Thatte' add+ verify-
```

(4) Additional Header Fixes

Some additional fixes need to be made to the header file before proceeding:

(a) Change the data & filetype of the DQF files.

They should be:

BITPIX = 16 /

DATATYPE= 'INTEGER*2' /

FILETYPE= DQF / shp, ext, edq, sdq, sci

But will originally be:

BITPIX = 32 /

DATATYPE= 'REAL*4' /

FILETYPE= BAS / shp, ext, edq, sdq, sci

Filetype can be set with hedit.

hedit superbias7.c1h FILETYPE DQF add+ verify

hedit superbias15.c1h FILETYPE DQF add+ verify-

(b) The datatype and bitpix are changed using the task chpixtype.

chpixtype superbias7.c1h newsuperbias7.c1h[1/4] short

chpixtype superbias7.c1h[2] newsuperbias7.c1h[2] short

chpixtype superbias7.c1h[3] newsuperbias7.c1h[3] short

chpixtype superbias7.c1h[4] newsuperbias7.c1h[4] short

!mv superbias7.c1h oldsuperbias7.c1h

!mv superbias7.c1d oldsuperbias7.c1d

!mv newsuperbias7.c1h superbias7.c1h

!mv newsuperbias7.c1d superbias7.c1d

chpixtype superbias15.c1h newsuperbias15.c1h[1/4] short

chpixtype superbias15.c1h[2] newsuperbias15.c1h[2] short

chpixtype superbias15.c1h[3] newsuperbias15.c1h[3] short

chpixtype superbias15.c1h[4] newsuperbias15.c1h[4] short

!mv superbias15.c1h oldsuperbias15.c1h

!mv superbias15.c1d oldsuperbias15.c1d

!mv newsuperbias15.c1h superbias15.c1h

!mv newsuperbias15.c1d superbias15.c1d

(c) Change the extensions (c0h -> r2h & c1h -> b2h).

!mv superbias7.c0h superbias7.r2h

!mv superbias7.c0d superbias7.r2d

!mv superbias7.c1h superbias7.b2h

```
!mv superbias7.c1d superbias7.b2d
```

```
!mv superbias15.c0h superbias15.r2h  
!mv superbias15.c0d superbias15.r2d  
!mv superbias15.c1h superbias15.b2h  
!mv superbias15.c1d superbias15.b2dp  
!mv superbias15.c0h superbias15.r2h  
!mv superbias15.c0d superbias15.r2d  
!mv superbias15.c1h superbias15.b2h  
!mv superbias15.c1d superbias15.b2d
```

(d) Update the GPIXELS keyword in both the .r2h and .b2h files to reflect the number of good pixels (obtained from running wstat on your superbias):

For Gain7

```
wstat superbias7.r2h usedqf=yes datextn=r2h dqfextn=b2h
```

```
# Image statistics for 'superbias7.r2h' with mask 'superbias7.b2h'  
# GROUP      NPIX      MIN      MAX      MEAN      MIDPT      MODE      STDDEV  
[  1 ] 639963 -0.0094152  81.0414  0.349846  0.34585  0.3379  0.17821  
[  2 ] 639958 -0.126834  114.402  0.328839  0.32422  0.32578  0.32154  
[  3 ] 639927 -0.0554886  404.422  0.342169  0.33391  0.33905  0.61453  
[  4 ] 639917 -0.17307  291.046  0.356152  0.34803  0.34691  0.47591
```

```
hedit superbias7.?2h[1] gpixels 639963 verify-  
hedit superbias7.?2h[2] gpixels 639958 verify-  
hedit superbias7.?2h[3] gpixels 639927 verify-  
hedit superbias7.?2h[4] gpixels 639917 verify-
```

For Gain 15

```
wstat superbias15.r2h usedqf=yes datextn=r2h dqfextn=b2h
```

```
# Image statistics for 'superbias15.r2h' with mask 'superbias15.b2h'  
# GROUP      NPIX      MIN      MAX      MEAN      MIDPT      MODE      STDDEV  
[  1 ] 639974 -0.0776141  41.2409  0.191397  0.1888  0.18623  0.098557  
[  2 ] 639953 -0.0810867  58.6738  0.167486  0.1644  0.16873  0.16545  
[  3 ] 639910 -0.0747086  202.592  0.193146  0.19086  0.20067  0.30903  
[  4 ] 639922 -0.0568256  151.216  0.20673  0.20256  0.19879  0.251
```

```
hedit superbias15.?2h[1] gpixels 639974 verify-  
hedit superbias15.?2h[2] gpixels 639953 verify-  
hedit superbias15.?2h[3] gpixels 639910 verify-  
hedit superbias15.?2h[4] gpixels 639922 verify-
```

(e) Change some keyword values (so fitsverify will work)

For gain7

```
hedit superbias7.?2h moonangl "" ver- update+
hedit superbias7.?2h fgslock "" ver- update+
hedit superbias7.?2h time-obs "" ver- update+
hedit superbias7.?2h expflag "" ver- update+
hedit superbias7.?2h seqname "" ver- update+
hedit superbias7.?2h date-obs '2007-05-14' ver- update+
```

For Gain15

```
hedit superbias15.?2h moonangl "" ver- update+
hedit superbias15.?2h fgslock "" ver- update+
hedit superbias15.?2h time-obs "" ver- update+
hedit superbias15.?2h expflag "" ver- update+
hedit superbias15.?2h seqname "" ver- update+
hedit superbias15.?2h date-obs '2007-05-07' ver- update+
```

III. CERTIFYING AND DELIVERING FILES

After you have created the superbias, they need to be delivered to the archive to be used in the pipeline calibration of WFPC2 data. Please follow CDBS TIR 2008-01 and 2008-02 for delivery quality checks and also check with CDBS (cdb@stsci.edu) if there are any changes from the procedure described below.

(1) Perform all Necessary Steps Described in the CDBS TIR

```
!source /grp/hst/cdb/tools/util/defines.csh
```

```
!/grp/hst/cdb/tools/bin/certify superbias7.?2h
!/grp/hst/cdb/tools/bin/mkload superbias7.?2h
```

```

#Update load files
!perl -pi -e 's/LEVEL = /LEVEL = SEVERE/g' *lod
!perl -pi -e 's/OPUS_FLAG = /OPUS_FLAG = Y/g' *lod

hssel (images="superbias7.r2h",fields="descrip",expr="yes",> "tmp")

!cat superbias7.r2h | grep HISTORY >> tmp

!perl -pi -e 's/HISTORY //g' tmp

!perl -ne 'print if(/FILE_NAME/ ? ($c=7) : (--$c>0))' superbias7.lod > junk

!sed "/COMMENT =/r tmp" junk > junk1

!perl -ne 'print if(/ENDHEADER/ ? ($c=13) : (--$c>0))' superbias7.lod > junk2

!cat junk1 junk2 > superbias7.lod

delete("junk*",ver-)
delete("tmp*",ver-)

!/grp/hst/cdbs/tools/bin/certify *.lod
!/grp/hst/cdbs/tools/bin/check_load *.lod

print superbias7.r2h > files_in
print superbias7.b2h >> files_in

print superbias7_r2f.fits > files_out
print superbias7_b2f.fits >> files_out

stwfits("@files_in","@files_out")
!farris_fitsverify *.fits
#Create unique filenames.
!/grp/hst/cdbs/tools/bin/uniqname superbias7*

```

(2) Make a linux Version of the Superbias

Create a directory 'linux' in the superbias working directory. Convert the GEIS superbias with the new name to waiver fits format using stwfits. Copy the waiver fits files and the .lod file to the linux directory. Ssh to a linux machine, start IRAF/PyRAF and go to the

linux directory. Convert the fits files to GEIS format using strfits. There is no need to do cdbS check on the linux version.

(3) Send e-mail

Send an e-mail to cdbs@stsci.edu (with a cc to biretta@stsci.edu and yourself) with the following template.

Subject: WFPC2 gain 7 Superbias delivery (Feb 2006-May 2007 yearly superbias)

To: cdbs@stsci.edu

1- Name of deliverer: Deepashri Thatte

2- Date of delivery: June 28, 2007

3- Instrument: WFPC2

4- Type of file (bias,pht,etc.): bias

5- History section in header [0] complete?: yes

6- USEAFTER, PEDIGREE, DESCRIP, and COMMENT have been checked?: yes

7- CDBS Verification complete? (farris_fitsverify, certify,etc.): Yes

8- Should these files be ingested in the OPUS, DADS and CDBS databases? Yes

9- Files run through CALXXX or SYNPHOT?: Yes (calwp2)

10- Does it replace an old reference file?: no

10a- If yes, which one?

11- What is the level of change of the file?: SEVERE

12- Description of how the files were "tested" for correctness:

Ran certify, calwp2, and compared the images with previous superbias.

13- Disk location and name of files:

Include both Solaris and Linux versions

Acknowledgements

Thanks to S. Gonzaga for checking the superbias files. Previous work by S. Gonzaga, M. Richardson and procedures by K. Rudloff were used in writing this report.

References

Biretta, J. and Gonzaga, S., "Early Assessment of the WF4 Anomaly,"
WFPC2 ISR 2005-02.

- Diaz, R. I. and Cracraft, M. "Assessment and Delivery of Reference Files,"
CDBS TIR 2008-01.
- Diaz, R. I. and Cracraft, M. "Delivery of Reference Files to the Data
Management Systems," CDBS TIR 2008-02.
- Richardson, M. "Creating a WFPC2 Yearly Superbias," WFPC2 TIR 2005-03.
- Richardson, M. and Biretta, J. "Creating a WFPC2 Yearly Superdark,"
WFPC2 TIR 2005-04.

Appendix A – Script fix4.cl for Correcting WF4 CCD Data

Here we list the IRAF script fix4.cl (version fix4bias_8jun2007.cl) used to correct WF4 anomaly in bias images.

```
-----  
procedure fix4(input)  
#  
#  
# This version was tailored to make the 2007 super-bias.  
# Changes:  
# - Gain correction applied bias >290 DN  
# - More diagnostics printed to streak_list file  
#  
# This script processes images from an input list,  
# makes linearity and gain corrections to WF4,  
# removes the streaks from WF4, and outputs images  
# with the same names into a subdirectory "fixed".  
#  
# A table listing the image names, their WF4 bias levels,  
# and WF4 streak intensities is also output to file "streak_list".  
#  
# A subdirectory "scratch" is created in the "fixed"  
# directory and used for temporary files. These temporary  
# files are not deleted after the final image is processed.  
#  
# The streak intensity is measured with the streaker.cl  
# algorithm.  
#  
# The only script input is the name of the file containing  
# list of the images to be processed.  
#  
# to run:  
#  
# generate fake param file  
# cp (some param file) test.par  
# mv test.par (to directory where running)  
# edit test.par to look plausible  
#  
# stsdas  
# hst  
# too  
# imgt  
# task fix4 = "fix4.cl"  
# fix4 junk < image_list  
#  
# Where "image_list" is a list of input images to  
# be processed, one per line. This can be generated by  
# ls *.c0h > image_list  
#  
# This version works only on 4-group GIES format WFPC2 images.  
#  
# A copy of the generated streak image is left in the  
# scratch directory as "streak.hhh".  
#
```

```

# JB 24nov2005
#

string input {prompt = "Name of input file?"}

begin
string expression, output2, name, junk, junk2
string resultsfile, out, scr, files, filename, filename4
real fval, upper, lower, streak_rms, biaseven
real arg1, lowercorr, gain, be
real l1, l2, l3, l4
real l_corr, inv_gain

# set and creat directories
out="fixed/"
scr="fixed/scratch/"
!mkdir fixed
!mkdir fixed/scratch

# set name of output text file
!rm fixed/streak_list
resultsfile=out// "streak_list"

# print results to file
files="          image          biaseven  streak RMS  lin_corr  inv_gain"
printf ("%47s\n",files)
printf ("%47s\n",files,
        >> resultsfile)

#
# loop through input file containing image names to process
while(scan(filename) != EOF){

# copy the input image to the scratch directory
imcop(filename//[1],scr//filename//[1/4])
imcop(filename//[2],scr//filename//[2])
imcop(filename//[3],scr//filename//[3])
imcop(filename//[4],scr//filename//[4])

# delete old scratch files
imdel(scr// "qq*.hhh")
imdel(scr// "streak.hhh")
#
# start processing wf4:
filename4=filename//[4]

# get BIASEVEN value
hedit (filename4,"BIASEVEN",".") | scan (junk,junk2,biaseven)
print (biaseven)

# compute linearity correction
# compute correction on bias level first
# linearity correction coeff
l1=4.
l2=30.
l3=20.
l4=290.

# only correct if biaseven < 290
l_corr=0.
if(biaseven < l4) {
imcop(filename4,scr// "qqa.hhh")
# the lower correction depends only on the bias level, so compute separately
arg1=(l4-biaseven)/l4
print (arg1)
lowercorr=l1*arg1+l2*arg1*arg1+l3*arg1*arg1*arg1

```

```

print (lowercorr)
# now compute the upper correction as function of (bias + pixel value)
# the net correction is (lower correction - upper correction)
expression="( (//l4//) - min((iml+//biaseven//"),(//l4//) ) )//l4
print (expression)
imcalc(scr// "qqa.hhh",scr// "qqarg2.hhh",expression)
expression=lowercorr/ "-( (//l1//"*iml+//l2//"*iml*iml+//l3//"*iml*iml*iml) "
print (expression)
imcalc(scr// "qqarg2.hhh",scr// "qqlincorr.hhh",expression)
# apply linearity correction
imarith(scr// "qqa.hhh", "+",scr// "qqlincorr.hhh",scr// "qqlinccorred.hhh")
# save and print the median correction
imstat(scr// "qqlincorr.hhh[100:750,100:750]",fields="mean",lower=-10,upper=10,
        nclip=3,lsigma=3.,usigma=3.,
        binwidht=0.1,format-,mode="ql") | scan (fval)
l_corr=fval

# biaseven > 290 skips linearity correction
} else {
imcop(filename4,scr// "qqlinccorred.hhh")
}

# compute gain correction
# gain correction is function of bias level only
be=biaseven
gain=0.848
gain=gain-0.000196*be
gain=gain+0.00000596*be*be
gain=gain-0.0000000320*be*be*be
gain=gain+0.000000000645*be*be*be*be
print (gain)
inv_gain=1./gain
# apply gain correction
imarith(scr// "qqlinccorred.hhh", "/",gain,scr// "qqcorrect.hhh")

# OLD VERSION SKIPPED GAIN CORRECTION (INCORRECTLY) FOR BIAS>290 DN
# -- EVEN AT BIAS 290 THERE IS A 3% CORRECTION TO MAKE
## biaseven > 290 skips corrections
#} else {
#imcop(filename4,scr// "qqcorrect.hhh")
#}

# remove streaks from image
#
# exclude edge of image
name=scr// "qqcorrect.hhh[3:798,3:798]"
print(name)
imcop(name,scr// "qq1.hhh")
# compute image median
imstat(scr// "qq1.hhh[100:750,100:750]",fields="mean",lower=-10,upper=10,
        nclip=3,lsigma=3.,usigma=3.,
        binwidht=0.1,format-,mode="ql") | scan (fval)
print(fval)
# set high and low regions to median
upper=fval+4.
lower=fval-4.
expression="if iml .gt. (//upper//) then (//fval//) else iml"
print (expression)
imcalc(scr// "qq1.hhh",scr// "qqup1.hhh",expression)
expression="if iml .lt. (//lower//) then (//fval//) else iml"
print (expression)
imcalc(scr// "qqup1.hhh",scr// "qqout1.hhh",expression)
# convolved with broad gaussian in X
gauss(scr// "qqout1.hhh",scr// "qqgauss1.hhh",100.,ratio=0.001,theta=0.,
      nsigma=3.,bilinea+,boundar="reflect")

```

```

# iterate #2
expression="if (im1-im2) .gt. 3. then im2 else im1"
print (expression)
files=scr// "qqout1.hhh",scr// "qqgauss1.hhh"
imcalc(files,scr// "qqlo2.hhh",expression)
expression="if (im2-im1) .gt. 3. then im2 else im1"
print (expression)
files=scr// "qqlo2.hhh",scr// "qqgauss1.hhh"
imcalc(files,scr// "qqout2.hhh",expression)
gauss(scr// "qqout2.hhh",scr// "qqgauss2.hhh",100.,ratio=0.001,theta=0.,
      nsigma=3.,bilinea+,boundar="reflect")

# iterate #3
expression="if (im1-im2) .gt. 2. then im2 else im1"
print (expression)
files=scr// "qqout2.hhh",scr// "qqgauss2.hhh"
imcalc(files,scr// "qqlo3.hhh",expression)
expression="if (im2-im1) .gt. 2. then im2 else im1"
print (expression)
files=scr// "qqlo3.hhh",scr// "qqgauss2.hhh"
imcalc(files,scr// "qqout3.hhh",expression)
gauss(scr// "qqout3.hhh",scr// "qqgauss3.hhh",100.,ratio=0.001,theta=0.,
      nsigma=3.,bilinea+,boundar="reflect")

# zero the streak image
imstat(scr// "qqgauss3.hhh"[100:750,100:750],fields="mean",lower=-10,upper=10,
      nclip=3,lsigma=3.,usigma=3.,
      binwid=0.1,format-,mode="ql") | scan (fval)
print(fval)
imarith(scr// "qqgauss3.hhh", "-", fval,scr// "qqgauss3x.hhh")

# generate results:

# repaired image
imarith(scr// "qq1.hhh", "-",scr// "qqgauss3x.hhh",scr// "qqxxx.hhh")
imcop(filename4,scr// "qqfinal.hhh")
imcop(scr// "qqxxx.hhh"[1:796,1:796],scr// "qqfinal.hhh"[3:798,3:798])
# final output image
imcop(scr//filename//[1],out//filename//[1/4])
imcop(scr//filename//[2],out//filename//[2])
imcop(scr//filename//[3],out//filename//[3])
imcop(scr// "qqfinal.hhh",out//filename//[4])

# streak image
imcop(filename4,scr// "streak.hhh")
imarith(scr// "streak.hhh", "*",0.,scr// "streak.hhh")
imcop(scr// "qqgauss3x.hhh"[1:796,1:796],scr// "streak.hhh"[3:798,3:798])

# stats on streak image
imstat(scr// "streak.hhh"[50:750,50:750],fields="stddev",lower=undef,upper=undef,
      nclip=0,lsigma=3.,usigma=3.,
      binwid=0.1,format-,mode="ql") | scan (fval)
print(fval)
streak_rms=fval

# print results to file
printf ("%20s %10.3f %10.3f %10.3f
%10.3f\n",filename4,biaseven,streak_rms,l_corr,inv_gain)
printf ("%20s %10.3f %10.3f %10.3f
%10.3f\n",filename4,biaseven,streak_rms,l_corr,inv_gain,
      >> resultsfile)

}

end

```

Appendix B – Displaying Images

To visually examine all 4 groups in all the images use the script `display_all.cl`:

```
-----  
task display_all=display_all.cl  
  
procedure display_all (images)  
  
# This program displays a list of images 'interactively'. The input should  
# be a list file with the names of the images to display WITHOUT SPECIFYING  
# THE CHIP/DETECTOR. To run this 'task' just write at the iraf prompt:  
# display_all listfiles  
#  
# Adjust the z1 and z2 values according to the images to be displayed.  
  
file images {prompt="List of images to display"}  
  
begin  
  
    int i  
    string fname  
  
    list = images  
  
    while (fscan(list,fname) != EOF) {  
        for (i=1; i<=4; i+=1)  
        {  
            display(fname//["/i/"],fill+,zs-,zr-,z1=-2,z2=5)  
#            display(fname//["/i/"],1,fill+,zs-,zr-,z1=-2,z2=5)  
            print(i)  
        }  
    }  
  
end  
  
-- Or use the following method to generate a script to display images.  
In the working directory  
#ls *c0h > listfiles.txt  
  
#The lines looks like  
#image1.c0h  
#image2.c0h  
#image3.c0h  
#  
# |  
# |  
# |  
#image134.c0h  
  
#Open listfiles.txt with emacs  
  
#Use 'Query replace' to replace 'u9' by 'display u9'. To replace all  
#occurances type 'shift !'  
  
#Make 4 copies of listfiles.txt  
  
#copy listfiles.txt display1.cl  
#copy listfiles.txt display2.cl  
#copy listfiles.txt display3.cl  
#copy listfiles.txt display4.cl  
  
#Open display1.cl with emacs  
#With 'Query Replace' replace '.c0h' with '.c0h[1]'
```

```
#Now display1.cl looks like

#display image1.c0h[1]
#display image2.c0h[1]
#display image3.c0h[1]
#
#
#
#display image134.c0h[1]

#Repeat for groups 2, 3 and 4

#cat display[1-4].cl >> display.cl
#This command appends all display files

#tsort display.cl c2
#This command sorts display.cl according to imagenames which are in column c2
```

Appendix C -- gnuplot Script

Use this gnuplot script to plot 'biaseven vs date' and biasodd vs date'.

File plot_biases.p

```
-----  
  
#File plot_biases.p is the gnuplot script to plot 'biaseven vs date'  
#and 'biasodd vs date' for chips 1 to 4 of WFPC2.  
#Type gnuplot at the prompt  
#load 'plot_biases.p'  
  
#set size square  
#set xrange [x1:x2]  
#set xtics (0.4,0.6,0.8,1.0,1.2,1.4)  
set yrange [315:325]  
set xlabel "Date"  
set ylabel "Biaseven"  
set term post enhance  
set output 'biaseven1_date.ps'  
plot "biaslused.txt" using 2:4 ti 'Biaseven vs Date for PC' with points  
pointsize 1  
  
#set xrange [x1:x2]  
#set xtics (0.4,0.6,0.8,1.0,1.2,1.4)  
set yrange [315:325]  
set xlabel "Date"  
set ylabel "Biasodd"  
set term post enhance  
set output 'biasodd1_date.ps'  
plot "biaslused.txt" using 2:3 ti 'Biasodd vs Date for PC' with points  
pointsize 1  
  
#set xrange [x1:x2]  
#set xtics (0.4,0.6,0.8,1.0,1.2,1.4)  
set yrange [350:360]  
set xlabel "Date"  
set ylabel "Biaseven"  
set term post enhance  
set output 'biaseven2_date.ps'  
plot "bias2used.txt" using 2:4 ti 'Biaseven vs Date for WF2' with points  
pointsize 1  
  
#set xrange [x1:x2]  
#set xtics (0.4,0.6,0.8,1.0,1.2,1.4)  
set yrange [350:360]  
set xlabel "Date"  
set ylabel "Biaseven"  
set term post enhance  
set output 'biasodd2_date.ps'  
plot "bias2used.txt" using 2:3 ti 'Biasodd vs Date for WF2' with points  
pointsize 1  
  
#set xrange [x1:x2]  
#set xtics (0.4,0.6,0.8,1.0,1.2,1.4)  
set yrange [305:315]  
set xlabel "Date"
```

```
set ylabel "Biaseven"  
set term post enhance  
set output 'biaseven3_date.ps'  
plot "bias3used.txt" using 2:4 ti 'Biaseven vs Date for WF3' with points  
pointsize 1
```

```
#set xrange [x1:x2]  
#set xtics (0.4,0.6,0.8,1.0,1.2,1.4)  
#set yrange [305:315]  
set xlabel "Date"  
set ylabel "Biasodd"  
set term post enhance  
set output 'biasodd3_date.ps'  
plot "bias3used.txt" using 2:3 ti 'Biasodd vs Date for WF3' with points  
pointsize 1
```

```
#set xrange [x1:x2]  
#set xtics (0.4,0.6,0.8,1.0,1.2,1.4)  
set yrange [200:350]  
set xlabel "Date"  
set ylabel "Biaseven"  
set term post enhance  
set output 'biaseven4_date.ps'  
plot "bias4used.txt" using 2:4 ti 'Biaseven vs Date for WF4' with points  
pointsize 1
```

```
#set xrange [x1:x2]  
#set xtics (0.4,0.6,0.8,1.0,1.2,1.4)  
set yrange [200:350]  
set xlabel "Date"  
set ylabel "Biasodd"  
set term post enhance  
set output 'biasodd4_date.ps'  
plot "bias4used.txt" using 2:3 ti 'Biasodd vs Date for WF4' with points  
pointsize 1
```

Appendix D -- dohedit2.cl

```
procedure
dohedit2(input_images,cdb_s_fil,atodgain,shutter,date,infiltnam1,infiltnam2,infilt1,infilt2
,inpfiltnam1,inpfiltnam2)

string input_images   {prompt="Input images (*.r?h)"}
string cdb_s_fil      {prompt="Input image type (MASK, ATOD, BIAS, SHAD, DARK,
FLAT)"}
real atodgain         {prompt="Input atodgain"}
string shutter        {prompt="Input shutter (A, B, or ' ')}
string date           {prompt="Input date (dd/mm/yy)"}
string infiltnam1     {prompt="Input filtnam1 (F439W)"}
string infiltnam2     {prompt="Input filtnam2"}
int infilt1           {prompt="Input filter1 (a number)"}
int infilt2           {prompt="Input filter2 (a number)"}
int inpfiltnam1       {prompt="Input pfilter1 (a number)"}
int inpfiltnam2       {prompt="Input pfilter2 (a number)"}

begin

string inimg, img, imgname
string imhing
string temp, tempimh
string shut, cdb_s_fil
string rootname, datemade
string filtnam1, filtnam2
string ext
int filt1,filt2,pfilt1,pfilt2
real min, max, atod
int i, j, l

    inimg = input_images; cdb_s_fil = cdb_s_fil
    if(cdb_s_fil != 'MASK'){if(cdb_s_fil != 'ATOD'){if(cdb_s_fil != 'BIAS'){
        if(cdb_s_fil != 'SHAD'){if(cdb_s_fil != 'DARK'){if(cdb_s_fil != 'FLAT'){
            print("cdb_s_file not an allowed value!");bye}}}}}}
    atod = atodgain
    if(atod != 15.0){if(atod != 7.0){if(atod !=0.0){print("atodgain not an allowed
value!");bye}}}
    shut = shutter
    if (shut != 'A'){if(shut != 'B'){if(cdb_s_fil == 'SHAD'){print"need a shutter value
A or B for SHAD files!");bye}}}
    datemade = date
    temp = mktemp("tmpfil")

    if (cdb_s_fil == 'FLAT') {
        filtnam1 = infiltnam1; filtnam2 = infiltnam2
        filt1 = infilt1; filt2 = infilt2
        pfilt1 = inpfiltnam1; pfilt2 = inpfiltnam2
    }

    print(" ")
    sections (inimg,option="fullname",mode="ql", >> temp)
    list = temp
    while (fscan(list,imgname) != EOF) {
        for(i=1; i<5; i+=1) {
            img = imgname+"["+i+"]"
            l = strlen(imgname)
            rootname = substr(imgname,1,l-4)
            ext = substr(imgname,l-2,l-2)
            minmax(img,force=yes,update=yes,verbose=no,mode="ql")
            min = minmax.minval
            max = minmax.maxval
            j = i

            hedit(img,"crval1",0.,add-,delete-,verify-,show-,update+,mode="ql")
            hedit(img,"crval2",0.,add-,delete-,verify-,show-,update+,mode="ql")
            hedit(img,"crval3",0.,add-,delete-,verify-,show-,update+,mode="ql")
            hedit(img,"crpix1",0.,add-,delete-,verify-,show-,update+,mode="ql")
            hedit(img,"crpix2",0.,add-,delete-,verify-,show-,update+,mode="ql")
        }
    }
end
```

```

hedit(img,"cdl_1",0.,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"cdl_2",0.,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"cd2_1",0.,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"cd2_2",0.,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"datamin",min,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"datamax",max,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"atodgain",atod,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"mir_rev", "F",add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"orientat",0.,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"fillcnt",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"errcnt",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"fpkctime", " " ,add-,delete-,verify-,show-,
update+,mode="ql")
hedit(img,"lpkctime", " " ,add-,delete-,verify-,show-,
update+,mode="ql")
hedit(img,"ctype1", " " ,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"ctype2", " " ,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"ctype3", " " ,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"detector",j,add-,delete-,verify-,show-,update+,mode="ql")

hedit(img,"dezero",0.,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"goodmin",min,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"goodmax",max,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"datamean",0.,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"biaseven",0.,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"biasodd",0.,add-,delete-,verify-,show-,update+,mode="ql")

hedit(img,"gpixels",640000,add-,delete-,verify-,show-,update+,mode="ql")
print("warning: setting GPIXELS to 640000")
hedit(img,"softerrs",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"calibdef",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"staticd",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"atodsat",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"datalost",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"badpixel",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"overlap",0,add-,delete-,verify-,show-,update+,mode="ql")

hedit(img,"photmode", " " ,add-,delete-,verify-,show-,
update+,mode="ql")
hedit(img,"photflam",0.,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"photzpt",0.,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"photplam",0.,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"photbw",0.,add-,delete-,verify-,show-,update+,mode="ql")

print("finished group ",i)
}
print(" ")
print("finishing rest of keywords")
img = imgname
hedit(img,"instrume", "WFPC2 " ,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"rootname",rootname,add-,delete-,verify-,show-,update+,mode="ql")
if (cdbsfil == 'MASK') {
hedit(img,"filetype", "MSK " ,add-,delete-,verify-,show-,update+,mode="ql")
}
if (cdbsfil == 'ATOD') {
hedit(img,"filetype", "A2D " ,add-,delete-,verify-,show-,update+,mode="ql")
}
if (cdbsfil == 'BIAS') {
hedit(img,"filetype", "BAS " ,add-,delete-,verify-,show-,update+,mode="ql")
}
}
hedit(img,"nsegpgrp",0,add-,delete-,verify-,show-,update+,mode="ql")

hedit(img,"mode", "FULL " ,add-,delete-,verify-,show-,update+,mode="ql")
print("warning: setting MODE to value FULL")
hedit(img,"serials", "OFF " ,add-,delete-,verify-,show-,update+,mode="ql")
print("warning: setting SERIALS to value OFF")

if (cdbsfil == 'MASK' || cdbsfil == 'ATOD' || cdbsfil == 'BIAS' || cdbsfil == 'DARK') {
hedit(img,"filtnam1", " " ,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"filtnam2", " " ,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"filter1",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"filter2",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"pfilter1",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"pfilter2",0,add-,delete-,verify-,show-,update+,mode="ql")
}

```

```

}

hedit(img, "exptime", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "uexpodur", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "darktime", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "uexpotim", 0.0000000E+00, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "nshutal7", 0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "pstrtime", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "pstptime", " ", add-, delete-, verify-, show-, update+, mode="ql")

hedit(img, "kspots", "OFF", add-, delete-, verify-, show-, update+, mode="ql")
print("warning: setting KSPOTS to value OFF")
if(cdbsfil != 'SHAD') {
hedit(img, "shutter", " ", add-, delete-, verify-, show+, update+, mode="ql")
}

hedit(img, "imagetyp", "CDBS", add-, delete-, verify-, show-, update+, mode="ql")
print("warning: setting IMAGETYP to value CDBS")
hedit(img, "cdbsfil", cdbsfil, add-, delete-, verify-, show+, update+, mode="ql")

hedit(img, "maskcorr", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "atodcorr", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "blevcorr", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "biascorr", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "darkcorr", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "flatcorr", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "shadcorr", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "dosatmap", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "dophotom", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "outdtype", " ", add-, delete-, verify-, show-, update+, mode="ql")

hedit(img, "maskfile", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "atodfile", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "blevfile", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "blevdfil", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "biasfile", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "biasdfil", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "darkfile", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "darkdfil", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "flatfile", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "flatdfil", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "shadfile", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "shaddfil", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "phottab", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "graphtab", " ", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "comptab", " ", add-, delete-, verify-, show-, update+, mode="ql")

hedit(img, "saturate", 4095, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "uscale", 1.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "uzero", 0.0, add-, delete-, verify-, show-, update+, mode="ql")

hedit(img, "readtime", 0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "uch1cjt", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "uch2cjt", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "uch3cjt", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "uch4cjt", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "ubay3tmp", 0.0, add-, delete-, verify-, show-, update+, mode="ql")

hedit(img, "pa_v3", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "ra_sun", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "dec_sun", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "eqnx_sun", 0.0E+00, add-, delete-, verify-, show-, update+, mode="ql")

hedit(img, "mtflag", "F", add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "egradtrg", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "flatntrg", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "npdectrg", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "npratrg", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "rotrttrg", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "longpmer", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "eplongpm", 0.0, add-, delete-, verify-, show-, update+, mode="ql")
hedit(img, "surflatd", 0.0, add-, delete-, verify-, show-, update+, mode="ql")

```

```

hedit(img,"pktfmt",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"surflong",0.0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"surfaltd",0.0,add-,delete-,verify-,show-,update+,mode="ql")

hedit(img,"podpsff",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"stdcfff",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"stdcfff",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"stdcfff",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"rsdpfill",-100,add-,delete-,verify-,show-,update+,mode="ql")

hedit(img,"date",datemade,add-,delete-,verify-,show+,update+,mode="ql")
hedit(img,"plotid",0,add-,delete-,verify-,show-,update+,mode="ql")

hedit(img,"proposid",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"pep_expo",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"targnaml",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"targnamc",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"plan_exp",0.0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"plan_sn",0.0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"priority",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"ra",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"dec",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"exposure",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"expstrt",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"expend",0,add-,delete-,verify-,show-,update+,mode="ql")

hedit(img,"sunangle",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"moonangl",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"fgslock",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"sun_alt",0.0E+00,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"date-obs",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"time-obs",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"expstart",0.0E+00,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"linenum",0,add-,delete-,verify-,show-,update+,mode="ql")
hedit(img,"seqline",0,add-,delete-,verify-,show-,update+,mode="ql")

if (cdbsfil == 'SHAD') {
  hedit(img,"exptime",1.0,add-,delete-,verify-,show+,update+,mode="ql")
  hedit(img,"filetype","SHD",add-,delete-,verify-,show+,update+,mode="ql")
  hedit(img,"shutter",shut,add-,delete-,verify-,show+,update+,mode="ql")
}

if (cdbsfil == 'FLAT') {
  hedit(img,"filtnam1",filtnam1,add-,delete-,verify-,show+,update+,mode="ql")
  hedit(img,"filtnam2",filtnam2,add-,delete-,verify-,show+,update+,mode="ql")
  hedit(img,"filter1",filt1,add-,delete-,verify-,show+,update+,mode="ql")
  hedit(img,"filter2",filt2,add-,delete-,verify-,show+,update+,mode="ql")
  hedit(img,"pfilter1",pfilt1,add-,delete-,verify-,show+,update+,mode="ql")
  hedit(img,"pfilter2",pfilt2,add-,delete-,verify-,show+,update+,mode="ql")
  hedit(img,"filetype","FLT",add-,delete-,verify-,show+,update+,mode="ql")
}

if (cdbsfil == 'DARK') {
  hedit(img,"darktime",1.0,add-,delete-,verify-,show+,update+,mode="ql")
  hedit(img,"filetype","DRK",add-,delete-,verify-,show+,update+,mode="ql")
}

if (ext == 'b') {
  hedit(img,"filetype","DQF",add-,delete-,verify-,show+,update+,mode="ql")
  # DON'T DELETE THE DOUBLE LINES HERE... IT JUST DIDN'T SEEM TO BE WORKING ONLY
  # DOING IT ONCE!!!
  hedit(img,"filetype","DQF",add-,delete-,verify-,show+,update+,mode="ql")
}
delete(temp)
print(" ")
end

```