Chandra X-ray Center

AHELP for CIAO 3.4　　　　　　　　　　**dmdiff**　　　　　　　　Context: tools

Jump to: Description Examples Parameters CHANGES IN CIAO 3.2 Bugs See Also

Synopsis

Compare values in two FITS files.

Syntax

```
dmdiff infile1 infile2 [outfile] [tolfile] [keys] [data] [subspaces]
[units] [comments] [wcs] [missing] [error_on_value] [error_on_comment]
[error_on_unit] [error_on_range] [error_on_datatype] [error_on_wcs]
[error_on_subspace] [error_on_missing] [verbose] [clobber] [mode]
```

Description

`dmdiff' requires two FITS files as input. The user can optionally enter a tolerance file and output file as well. The input filenames may be entered with an extension (i.e. inputfile.fits[EVENTS]) or any other virtual filters supported by the Data Model (DM). The "keys" parameter determines if the header keys will be checked, and the "data" parameter determines if the table/image data will be checked.

Exit Status

Like the Unix commands `diff' and `cmp', dmdiff assigns special meaning to its exit status. An exit status of 0 means that no differences were found in the two input files. An exit status of 1 means that either differences were found or an error occurred. An exit status greater than one always indicates that an error occurred. Note that if the verbose parameter is set to 0, the tool will produce no output, but the exit status will still reflect whether differences were found in the input files. This feature can be useful in scripts that automatically compare a large number of files.

Tolerance Files

The purpose of the tolerance file is to set parameters when comparing the values of the input files. There are many options for setting limits on the values within these files: [1] Set a minimum and maximum value for the keyword/column on a single line. For example, ccd_id=6:8 gives a ccd range from 6 to 8. [2] Set a minimum value for the keyword/column. For example, ccd_id=5: sets a minimum ccd value of 5. [3] Set a maximum value for the keyword/column. For example, ccd_id=:4 sets a maximum ccd value of 4. [4] Set a range for the difference allowed between the values in the two files. For example, chipx=range(1) would allow chipx values of the same table row to differ by no more than 1. [5] Set a value for the keyword/column. For example, ccd_id=8 only allows for the ccd to be chip 8. [6] Set a percent range for the keyword/column. For example, pha=%1 would allow pha values to differ by no more than 1 percent. [7] Ignore the directory path of a keyword value. For example, gainfile=ignorepath will compare the gainfile strings without the preceding

paths.

Current Limitations

There are a few limitations in the tool:

- The number of columns in the two infiles must be the same. If the number of rows are different, dmdiff will only compare the lesser number of rows in the inputs (i.e. infile1=1000 rows, infile2=500 rows; first 500 rows of each file are compared).
- Columns with the same name in the input files must also have the same datatype. dmdiff cannot compare columns of different types and will exit with an error if there is a mismatch.

Example 1

```
dmdiff dmdiff1.fits dmdiff2.fits
```

Compare all header and block values in dmdiff1.fits and dmdiff2.fits.

Example 2

```
dmdiff infile1=dmdiff1.fits[EVENTS] infile2=dmdiff2.fits[EVENTS]
```

Compare all the header and table values of the EVENTS block in dmdiff1.fits and dmdiff2.fits.

Example 3

```
dmdiff infile1=dmdiff1.fits[EVENTS] infile2=dmdiff2.fits[EVENTS]
keys=yes data=no
```

Compare only the header values of the EVENTS block in dmdiff1.fits and dmdiff2.fits.

Example 4

```
dmdiff infile1=dmdiff1.fits[2] infile2=dmdiff2.fits[2]
tolfile=tolerances.txt outfile=diffs.txt
```

Compare the header and table/image values listed in block 2 of dmdiff1.fits and dmdiff2.fits using the columns or keywords listed in tolerances.txt. Output will be written to diffs.txt.

Parameters

name	type	ftype	def	min	max	reqd	stacks
infile1	file	input				yes	no
infile2	file	input				yes	no
outfile	file					no	no
tolfile	file					no	no
keys	boolean		yes			no	

data	boolean		yes			no	
subspaces	boolean		yes			no	
units	boolean		yes			no	
comments	boolean		yes			no	
wcs	boolean		yes			no	
missing	boolean		yes			no	
error_on_value	boolean		yes			no	
error_on_comment	boolean		yes			no	
error_on_unit	boolean		yes			no	
error_on_range	boolean		yes			no	
error_on_datatype	boolean		yes			no	
error_on_wcs	boolean		yes			no	
error_on_subspace	boolean		yes			no	
error_on_missing	boolean		yes			no	
verbose	integer		1	0	5	no	
clobber	boolean		no			no	

Detailed Parameter Descriptions

Parameter=infile1 `(file required filetype=input stacks=no)`

The first input to use in file comparison.

Parameter=infile2 `(file required filetype=input stacks=no)`

The second input to use in file comparison.

Parameter=outfile `(file not required stacks=no)`

Output file listing summary of differences found in two fits files. If the value is omitted or set to 'none', 'NONE', or 'stdout', output will go to the standard output device (generally the terminal). If outfile is set to 'stderr', output will go to the standard error device (also generally displayed on the terminal). Finally, if a filename is given, output will be written to that file. The clobber parameter controls whether an existing file will be overwritten.

Parameter=tolfile `(file not required stacks=no)`

Ascii text tolerance file listing the header keywords and column names that are to be compared with parameters, and the keys/columns that are to be ignored. The ranges, minimum and maximum values for each keyword and column are listed as well, in the following format:

TSTART=83201992:83201992.7

chipx=range(50)

#time=83202418:

ccd_id=8

!checksum

!datasum

Keywords are not case sensitive. A leading '#' signifies the line as a comment. A leading '!' means that the keyword (or column) should be ignored. The first line provides a minimum and maximum value for TSTART, seperated by the ':'. The second line signifies that the chipx values in the two files can not differ by more than 50. The third line is a comment, but would provide a minimum value for the TIME column, if used. The fourth line signifies that the CCD_ID column must be equal to 8. The fifth and sixth lines tell the program not to include the CHECKSUM and DATASUM keywords, respectively.

Parameter=keys `(boolean not required default=yes)`

Determines whether or not the header keys will be checked.

Parameter=data `(boolean not required default=yes)`

Determines whether or not the table/image data will be checked.

Parameter=subspaces `(boolean not required default=yes)`

Determines whether or not the subspaces will be checked.

Parameter=units `(boolean not required default=yes)`

Determines whether or not the units will be checked.

Parameter=comments `(boolean not required default=yes)`

Determines whether or not the comments will be checked.

Parameter=wcs `(boolean not required default=yes)`

Determines whether or not the wcs will be checked.

Parameter=missing `(boolean not required default=yes)`

Determines if missing header keys will be checked.

Parameter=error_on_value `(boolean not required default=yes)`

Return error when values are different?

Parameter=error_on_comment `(boolean not required default=yes)`

Return error when comments are different?

Parameter=error_on_unit `(boolean not required default=yes)`

Return error when units are different?

Parameter=error_on_range `(boolean not required default=yes)`

Return error when ranges are different?

Parameter=error_on_datatype `(boolean not required default=yes)`

Return error when datatypes are different?

Parameter=error_on_wcs `(boolean not required default=yes)`

Return error when wcs's are different?

Parameter=error_on_subspace `(boolean not required default=yes)`

Return error when subspaces are different?

Parameter=error_on_missing `(boolean not required default=yes)`

Return error when header key is missing?

Parameter=verbose `(integer not required default=1 min=0 max=5)`

Verbosity level of terminal display information to user (DataModel output included). If verbose is set to 0, the tool will produce no output, but its exit status will indicate whether differences were found in the input files. See the section "Exit Status" above.

Parameter=clobber `(boolean not required default=no)`

Clobber existing output with the same name?

CHANGES IN CIAO 3.2

The tool now checks:

- images (1D and 2D)
- WCS
- DSS
- Units
- Comments
- Datatypes
- Missing keys
- Ranges

Bugs

See the bugs page for this tool on the CIAO website for an up–to–date listing of known bugs.

See Also

tools

dmhedit, dmhistory, dmkeypar, dmmakepar, dmreadpar

Parameter=error_on_range (boolean not required default=yes)