



AHELP for CIAO 3.4

dmhistory

Context: [tools](#)*Jump to:* [Description](#) [Examples](#) [Parameters](#) [Bugs](#) [See Also](#)

Synopsis

Extracts the tool history from the file header

Syntax

```
dmhistory infile tool [outfile] [expand] [action] [clobber] [verbose]
```

Description

'dmhistory' extracts the processing history stored in a data file's HISTORY keywords. It is possible to extract individual commands by tool name or to retrieve the entire file history (multiple tools) in one batch.

The HISTORY keywords are a part of the COMMENTS header section of a data file and contain a listing of all the commands (tools plus their input parameters) which were used to process the file. To view the actual history, type: `dmlist file.fits opt=comments`.

Example 1

```
dmhistory file.fits dmhedit
```

Will extract the history records from 'file.fits' and print to STDOUT all dmhedit commands that were used.

Example 2

```
dmhistory file.fits acis_format_events expand=yes
```

Will extract the history records from 'file.fits' and print to STDOUT any acis_format_events commands that were used. Will also expand the stacks in the output:

```
acis_format_events infile="filename.fits[time=51517057.4305350:51522856.8805350]" ....
```

Otherwise, the output would be abbreviated as such:

```
acis_format_events infile="@filename.fits" ....
```

Example 3

```
unix% punlearn dmcop
unix% dmhistory file.fits dmcop action=pset
unix% plist dmcop
```

This will grab the 'dmcop' history record from the file file.fits and use it to set the parameters in dmcop.par. If multiple dmcop commands were run, the last instance of dmcop is reflected in the .par file.

Example 4

```
dmhistory file.fits my_own_script action=put
```

Read the parameters from the 'my_own_script.par' parameter file and add it to the history of 'file.fits'. 'my_own_script.par' will be searched for in the usual parameter searching order.

Parameters

name	type	ftype	def	min	max	reqd
<u>infile</u>	file	input				yes
<u>tool</u>	string					yes
<u>outfile</u>	file	output	stdout			
<u>expand</u>	boolean		yes			
<u>action</u>	string		get			
<u>clobber</u>	boolean		no			
<u>verbose</u>	integer		0	0	5	

Detailed Parameter Descriptions

Parameter=infile (file required filetype=input)

The input file name (image or table)

Parameter=tool (string required)

The name of the tool whose history you wish to extract.

All the history for the specified tool will be extracted. For example, if dmhedit was run several times, you will extract all the dmhedit commands. You can specify 'all' to extract the history for all CIAO tools recorded in the file.

Parameter=outfile (file filetype=output default=stdout)

The output file name

The default will print one command per line to stdout. Optionally, a file name can be supplied. "none" or "" will direct the output to stdout as well.

Parameter=expand (boolean default=yes)

Expand stacks

This parameter controls whether or not the stack name or the list of files in the stack are included in the output string. For example, if 'file.lis' contained 'one' and 'two', setting expand=no will show '@file.lis' while expand=yes will show 'file.lis[one,two]'.

Parameter=action (string default=get)

What to do with the history: get, put, or pset

Defines what action dmhistory should take with the history information.

action="get": the command line string for 'tool' will be created and sent to the 'outfile' (default is stdout, i.e. the screen).

action="pset": instead of printing the value to 'outfile', the reconstructed command is "pset" (see "ahelp pset"), setting the parameter file for the 'tool' task. Note: If there are multiple instances of that tool in the history, then there will be multiple psets and the most recent instance will be the final version in the parameter file.

action="put": the parameter file for the given tool is written to the data file, instead of a command being read from the history. The infile must have read and write permission for the user. This feature is especially useful in shell–scripts that use a pseudo parameter interface (e.g. wavdetect and tgdetect) to store their parameters in the history. In "put" mode, 'all' is not a valid tool name unless the user is trying to add a command stored in 'all.par'. Also, the outfile and clobber parameters are not used in this mode.

Parameter=clobber (boolean default=no)

Remove output if it exists?

Used to specify whether or not to overwrite existing file that has the same name as the specified output file (only DM dataset will be clobbered).

Parameter=verbose (integer default=0 min=0 max=5)

Verbose can be from 0 to 5, generating different amounts of output.

Bugs

See the [bugs page for this tool](#) on the CIAO website for an up–to–date listing of known bugs.

See Also

concept

[parameter](#)

modules

[paramio](#)

paramio

[paccess](#), [paramclose](#), [paramopen](#), [pget](#), [pgets](#), [plist_names](#), [pquery](#), [pset](#), [punlearn](#)

tools

[dmdiff](#), [dmhedit](#), [dmkeypar](#), [dmmakepar](#), [dmreadpar](#), [paccess](#), [pdump](#), [pget](#), [pline](#), [plist](#), [pquery](#), [pset](#), [punlearn](#)

The Chandra X-Ray Center (CXC) is operated for NASA by the Smithsonian Astrophysical Observatory.
60 Garden Street, Cambridge, MA 02138 USA.
Smithsonian Institution, Copyright © 1998–2006. All rights reserved.

URL:
<http://cxc.harvard.edu/ciao3.4/dmhistory.html>
Last modified: December 2006