Chandra X-ray Center

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

Jump to: Description Examples Parameters Bugs See Also

Synopsis

stk_build is used to build a stack from a user–defined string. The output is to stdout, stderr, or a file.

Syntax

```
stk_build infile outfile [clobber] [verbose] [mode]
```

Description

The stk_build program takes an infile string and a user–defined outfile as input. The infile string can be defined in any variation that is accepted by the ASCDS stack API. These variations include the use of @, added directory paths, and blocks (ie. @/tmp/foo[PRIMARY]).

If the outfile parameter is defined to be stdout, all ouput will be directed to stdout. Similarly, if outfile is defined as stderr, all output will be to stderr. Any other name entered for outfile will be interpreted as a filename.

If the user wishes to control the write protection of the outfile from within the program, the clobber parameter can be set to "yes" or "no". The level of verbosity is defined by the verbose parameter. The user can optionally set verbose from 0 (for not output) up to 3 (for the most output.)

Example 1

```
stk_build infile="one two three" outfile=out.lis
```

Build a stack with the infile string "one two three". Output the stack's contents to out.lis.

Example 2

```
stk_build infile="@file.lis foo" outfile=stderr
```

Build a stack with the infile string "@file.lis foo". Output the stack's contents to stderr.

Example 3

```
stk_build infile="@/tmp/file.lis foo" outfile=stdout
```

Build a stack with the infile string "@/tmp/file.lis foo". Output the stack's contents to stdout.

Parameters

name	type	ftype	def	min	max	reqd	stacks
infile	file	input				yes	yes
outfile	file	output				yes	
clobber	boolean		no			no	
verbose	integer		0	0	5	no	

Detailed Parameter Descriptions

Parameter=infile `(file required filetype=input stacks=yes)`

The string used to build a stack.

The infile string can be defined in any variation that is accepted by the ASCDS stack API. These variations include the use of @, added directory paths, and blocks (ie. @/tmp/foo[PRIMARY]). Single or multiple stacks provided as input will always produce a single output file, or a single list to stdout/stderr.

Parameter=outfile `(file required filetype=output)`

Output to stdout, stderr, or a file name.

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

Regulates whether or not stk_build can overwrite an output file.

Set to "yes" if the outfile can be overwritten, "no" if it can not.

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

Determines the level of output the user will see when the program is run.

Each level adds additional output. 0 – no output. 1 – shows non–hidden parameters. 2 – shows hidden parameters. 3 – gives the user information about which stage of the program is executing.

Bugs

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

See Also

concept
      stack

Example 3

modules

stackio

tools

stk_count, stk_read_num, stk_where

Example 3   3

Example 3