URL: http://cxc.harvard.edu/ciao3.4/stackio.html
Last modified: December 2006

AHELP for CIAO 3.4

stackio

Context: modules

Jump to: Description Examples EXAMPLE OF USING THE STACKIO MODULE CHANGES IN CIAO 3.2 Bugs See Also

Synopsis

The S−Lang interface to the CXC stack library

Description

The stackio module is the interface between the S−Lang interpreter (see "ahelp slang") and the CXC stack library. Briefly, a stack consists of a list of strings stored in a text file (e.g. stk.lis), or a single string. In the former case, the stack is referred to via the @stk.lis convention. This will return an N−element stack, one item for each line in the input file. The latter case of a simple string, however, is also a legal stack, with a single element equal to the input string. See "ahelp stack" for more information.

This document provides an overview of the features of the stack module, and tips for using it efficiently in a S−Lang program. Detailed descriptions of each function are provided by individual ahelp pages.

The stackio module is not available by default; to use it in a S−Lang program, it must be loaded using the S−Lang require() function:

```
  require("stackio");
```

Functions provided by the module

The following functions are provided by the module; use "ahelp <function>" to get a detailed description of a function:

Function name with arguments
Stack_Type stk_build(String_Type,[Integer_Type])
Void_type stk_close(Stack_Type)
Integer_Type stk_count(Stack_Type)
Integer_Type stk_current(Stack_Type)
Integer_Type stk_set_current(Stack_Type, Integer_Type)
String_Type stk_read_next(Stack_Type)
Void_Type stk_rewind(Stack_Type)

Integer_Type stk_delete_current(Stack_Type)
Void_Type stk_disp(Stack_Type)
Integer_Type stk_delete_num(Stack_type, Integer_Type)
String_Type stk_read_num(Stack_Type, Integer_Type)
Integer_Type stk_append(Stack_Type, String_Type, [Integer_Type])
Integer_Type stk_change_current(Stack_type, String_Type)
Integer_Type stk_change_num(Stack_Type, String_Type, Integer_Type)
Stack_Type stk_expand_n(String_Type, Integer_Type)

Using Stack_Type variables

The stackio module defines a new variable type, Stack_Type, which has a complex internal structure that cannot be printed using the S−lang print() command. To see the components of a Stack_Type variable, use the stk_disp() call. Stack_Type variables can be defined using the stk_build() function, which converts a CIAO stack string into a Stack_Type variable.

Example 1

```
chips> require("stackio")
chips> stk = stk_build("acis_evt2.fits")
chips> stk_disp(stk)
−−−−−−
Stack position: 0
Stack size: 1
Stack allocated: 100
Stack entries:
1 :acis_evt2.fits:
−−−−−−
chips> () = stk_append(stk,"hrc_leg−1.fits")
chips> () = stk_append(stk,"hrc_leg1.fits")
chips> stk_read_num(stk,2)
hrc_leg−1.fits
chips> stk_delete_num(stk,2)
chips> stk_count(stk)
2
chips> stk_close(stk)
```

Example 2

```
sherpa> require("stackio")
sherpa> stk = stk_expand_n("acisf00#.fits",4)
sherpa> stk_disp(stk)
−−−−−−
Stack position: 0
Stack size: 4
Stack allocated: 4
Stack entries:
1 :acisf001.fits:
2 :acisf002.fits:
3 :acisf003.fits:
4 :acisf004.fits:
```

```
------
sherpa> stk_read_next(stk)
acisf001.fits
sherpa> () = stk_change_num(stk,"hrc_evt2.fits",2)
sherpa> stk_current(stk)
1
sherpa> () = stk_delete_current(stk)
sherpa> stk_append(stk,"@/data/ciao/stack.lis",1)
sherpa> stk_disp(stk)
------
Stack position: 1
Stack size: 6
Stack allocated: 6
Stack entries:
1 :hrc_evt2.fits:
2 :acisf003.fits:
3 :acisf004.fits:
4 :/data/ciao/a.dat:
5 :/data/ciao/b.dat:
6 :/data/ciao/c.dat:
------
sherpa> stk_close(stk)
```

EXAMPLE OF USING THE STACKIO MODULE

The following routine can be used to convert a Stack_Type variable into an array of strings, which may be easier to use in S−lang.

```
  %
  % Usage:
  %  result = stk_to_array( stack );
  %
  require("stackio");
  define stk_to_array ( stack ) {

    if (typeof(stack) != Stack_Type) {
      message("Error: Input not a stack.");
      return NULL;
    }

    variable n = stk_count( stack );
    variable result = String_Type [n];

    _for ( 0, n−1, 1 ) {
      variable i = ();
      result[i] = stk_read_num( stack, i+1 );
    }
    return result;
  } % stk_to_array( stack )
```

With the above function, a stack can be easily converted to a string array, as demonstrated below.

```
sherpa> evalfile("stk_to_array.sl");
1
sherpa> stk = stk_expand_n("acisf0812N00#_evt2.fits",3);
sherpa> stk_disp(stk)
------
Stack position: 0
Stack size: 3
Stack allocated: 3
Stack entries:
1 :acisf0812N001_evt2.fits:
2 :acisf0812N002_evt2.fits:
3 :acisf0812N003_evt2.fits:
−−−−−−
sherpa> array = stk_to_array(stk)
sherpa> print(length(array))
3
sherpa> print(array)
acisf0812N001_evt2.fits
acisf0812N002_evt2.fits
acisf0812N003_evt2.fits
```

CHANGES IN CIAO 3.2

The module can now be loaded by using the

```
require("stackio");
```

statement, although the previous method (loading with the import command) still works.

The stk_build() routine now returns a NULL when given a stack like "@foo" and the file foo does not exist. The stk_read_next() routine will now return NULL when called at the end of a stack rather than cause an "Intrinsic Error".

Bugs

See the bugs page for the stackio library on the CIAO website for an up−to−date listing of known bugs.

See Also

calibration
: caldb

chandra
: coords, guide, isis, level, pileup, times

chips
: chips

concept
: autoname, parameter, stack, subspace

dm
: dm, dmbinning, dmcols, dmfiltering, dmimages, dmimfiltering, dmintro, dmopt, dmregions, dmsyntax

gui
: gui

modules
: paramio, pixlib

slang
: overview, slang, tips

stackio
: stk_append, stk_build, stk_change_current, stk_change_num, stk_close, stk_count, stk_current, stk_delete_current, stk_delete_num, stk_disp, stk_expand_n, stk_read_next, stk_read_num, stk_rewind, stk_set_current

tools

stk_build, stk_count, stk_read_num, stk_where

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/stackio.html
Last modified: December 2006