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

AHELP for CIAO 3.4

stk_build

Context: stackio

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

Synopsis

Build a stack from text input (file name or regular expression).

Syntax

```
Stack_Type stk_build( String_Type stackstring )
Stack_Type stk_build( String_Type stackstring, Integer_Type prepend )
```

Description

This function converts a string (stackstring) into a stack. The stackstring argument can either be a single string or a filename preceeded by an @ sign. The former case creates a single−element stack whose only value is the stackstring unless stackstring is a pathname that includes a wildcard such as *, in which case all the matching filenames in that directory are used. In the case where an stack file is given with an @ sign, the file is read and a stack created with one element for each line in the file. The return value will be set to NULL on error.

The optional prepend argument determines whether the full path should be prepended to each element of the stack; 1 for yes and 0 for no. If not given it default to 1. This functionality allows you to access a group of files present in a separate directory using a stack file also present in that directory.

Example 1

```
chips> stk = stk_build( "a.fits,b.fits, c.fits" )
chips> stk_disp( stk )
------
Stack position: 0
Stack size: 3
Stack allocated: 100
Stack entries:
1 :a.fits:
2 :b.fits:
3 :c.fits:
------
```

The stk_build() routine can take in a string − such as would be given as inout for a tool parameter value − and convert it into a stack. Here we have used a list of files, separated by the "," character, and stk_build() has returned a stack of these elements.

The stk_count(), stk_read_next(), and stk_read_num() routines can be used to examine the contents of the stack.

Example 2

```
chips> asolstack = stk_build( "primary/pcad*asol1.fits" )
chips> stk_count( asolstack )
2
```

Here we have created a stack that contains all the files that match the expression

```
primary/pcad*asol1.fits
```

The stk_count() routine shows that there are two such files in this example.

Example 3

```
chips> stk = stk_build( "@stack.lis" )
chips> stk_disp(stk)
------
Stack position: 0
Stack size: 3
Stack allocated: 100
Stack entries:
1 :a.dat:
2 :b.dat:
3 :c.dat:
------
```

In this example the stack is created using the text file stack.lis, whose contents are:

```
a.dat
b.dat
c.dat
```

Example 4

```
chips> stk = stk_build( "@data/stack.lis" )
chips> stk_disp(stk)
------
Stack position: 0
Stack size: 3
Stack allocated: 100
Stack entries:
1 :data/a.dat:
2 :data/b.dat:
3 :data/c.dat:
------
```

Stacks can also be read from files in different directories. In this example the stack is created using the text file data/stack.lis, whose contents are:

```
a.dat
b.dat
c.dat
```

Since the optional second argument was not given, it defaulted to 1, which means that the path to the stack file is prepended to each element of the stack.

Example 5

```
chips> stk = stk_build( "@data/stack.lis", 0 )
chips> stk_disp(stk)
------
Stack position: 0
Stack size: 3
Stack allocated: 100
Stack entries:
1 :a.dat:
2 :b.dat:
3 :c.dat:
------
```

To avoid the path being added to the elements of the stack − as occurred in the previous example − stk_build() should be called with the second argument set to 0.

Example 6

```
chips> phastack = stk_build( "*.pha" )
chips> stk_count( phastack )
1
chips> stk_disp( phastack )
------
Stack position: 0
Stack size: 1
Stack allocated: 100
Stack entries:
1 :*.pha:
------
chips> stk_read_num( phastack, 1 )
*.pha
```

In this example there are no files that match the pattern

```
*.pha
```

In this situation. stk_build() returns a stack which contains one element, the contents of which match the supplied pattern.

CHANGES IN CIAO 3.2

The routine now returns NULL if given a stackstring like "@foo" and the file foo does not exist.

Bugs

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

See Also

modules
    stackio
stackio

stk_append, 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