Chandra X-ray Center

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

AHELP for CIAO 3.4

varmm

Context: modules

Jump to: Description CHANGES IN CIAO 3.2 CHANGES IN CIAO 3.1 CHANGES IN CIAO 3.0 Bugs See Also

Synopsis

The Variable, Math and Macro S−Lang library

Description

This document describes the functions provided by the Varmm library (Variable, Math and Macro) and how they can be used in S−Lang code. Additional useful information can be found using "ahelp slang", "ahelp variables", "ahelp math", and "ahelp tips".

The Varmm library is automatically loaded into Sherpa and ChIPS and can be loaded into any S−Lang code by saying

```
require("varmm");
```

Note that if you have already loaded the Sherpa or ChIPS S−Lang modules into your S−Lang code then you do not need to load in Varmm as well.

Varmm provides a number of functions for reading data from files (both ASCII and FITS formats), and assigning them to variables. In addition to Varmm, there are many I/O functions within S−Lang itself which can also be used; these functions resemble their C counterparts, such as fopen() and fread(). Varmm also contains a number of utility functions; use "ahelp <function>" to get a detailed description of a particular function or "ahelp −c varmm" to get a list of all these functions.

Function name with arguments
apropos()
clearstack()
dup_struct(s [,selectedfields])
fits_bitpix([data])
is_struct_defined(sf)
print(v)
printarr(arr[,n])
readarf(filename)
readascii(filename, [cols,nskip])
readbintab(filename, [cols,nskip])
readfile(filename, [cols,nskip])

readimage(filename)
readpha(filename)
readrdb(filename, [cols,nskip])
readrmf(filename)
reverse(a)
set_state(name, structure)
set_state_defaults(name)
username()
writeascii(filename, a, b, ...)
__exit(exitcode)

Customizing the S−Lang run−time environment

The Varmm library allows customization of your S−Lang run−time environment through the use of a resource file ($HOME/.varmmrc, discussed below) and the varmmrl module, described in "ahelp varmmrl".

If the file .varmmrc exists in your home directory, then the Varmm library will assume it contains a series of valid S−Lang commands and read it using evalfile(). This is a simple but powerful extension that permits essentially arbitrary S−Lang scripts/functions to be shared amongst any Varmm−enabled application (currently ChIPS and Sherpa). For example, if the following were stored in your $HOME/.varmmrc

```
  define compute_average (x, y)
  {
      variable s = x + y;
      return s / 2.0;
  }
```

then the function compute_average() would be available to each instance of Sherpa and ChIPS. The use of this feature should probably be restricted to often−used code, since it is read in and parsed every time a Varmm−aware CIAO application is started.

The Varmm state object

Another way that Varmm can be customized is through the Varmm state object. The varmm object contains a number of fields that can be used to change the default behaviour of Varmm functions (for instance by setting them in your $HOME/.varmmrc file).

The Varmm state object

Parameter name	Default value
readheader	1
rawkeys	1
verbosity	1
caseinsen	0

Currently only the readheader and caseinsen parameters are used, as discussed in "ahelp readfile". To see the current settings, you can use print and manipulate individual fields:

```
  chips> print(varmm)
  readheader      =  1
  rawkeys         =  1
  verbosity       =  1
  caseinsen       =  0
  chips> varmm.caseinsen = 1
```

and the set_state_defaults() and set_state() commands can be used to make wholesale changes to the object.

Using Varmm in your own S−Lang code

The following is not necessary if you are executing S−Lang code from ChIPS or Sherpa, since they already load the Varmm library, but is required if you are using a different S−Lang interpreter, such as slsh. The Varmm library can be made available by using the S−Lang "require()" command in your script:

```
  require("varmm");
```

You do not need to do this if the script also uses ChIPS or Sherpa S−Lang functions, since Varmm is automatically loaded when you do

```
require("chips");
```

or

```
require("sherpa");
```

CHANGES IN CIAO 3.2

The module can now be loaded by using the

```
require("varmm");
```

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

CHANGES IN CIAO 3.1

Improvements in image handling

The readimage command no longer caches images, so you can now read in data from a file if it has changed on disk (this cacheing was removed for table reading in the CIAO 3.0 release). The writefits() command now correctly writes out non−square images.

Reading in tables is now faster

Reading in FITS tables which do not contain array columns is now significantly faster.

cumsum() can now be used from ChIPS and Sherpa

The cumsum() function − part of the S−Lang Run−Time Library − can now be used from ChIPS and Sherpa.

The print() function on OS−X

The print() function can now be used on OS−X machines. In CIAO 3.0 the function tried to use /bin/more if the PAGER environment variable was not set; it now uses /usr/bin/more on OS−X machines.

CHANGES IN CIAO 3.0

import("varmm");

It is now possible to load the Varmm module − using import("varmm") − into S−Lang programs which are executed by slsh. In earlier versions of CIAO it was only available by loading ChIPS.

RDB support

The readrdb() function has been added to read in data stored in RDB files.

Addition of _filetype field

The structure returned by the read series of functions now contains the '_filetype' field, which defines the type of the input file. A number of constants (such as ASCII and RDB) are available to help interpret the contents of this field.

New functions

The apropos() function, whilst available in earlier versions of CIAO, is now documented. The fits_bitpix() routine has been added.

Deprecated functions

The min(), max(), and sum() functions are now part of the S−Lang Run−Time Library so the Varmm functions are deprecated. Since the S−Lang "RTL" verions have the same interface as the Varmm functions users will see no difference due to this change.

Bugs

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

See Also

chips
: chips, chips_eval

concept
: configure

gui
: analysis−menu, gui

sherpa
: sherpa_eval

slang
: math, overview, slang, tips, variables

tools
: ascii2fits

varmm
: __exit, apropos, clearstack, dup_struct, fits_bitpix, is_struct_defined, print, readarf, readascii, readbintab, readfile, readimage, readpha, readrdb, readrmf, reverse, set_state, set_state_defaults, username, writeascii, writefits

The Chandra X−Ray Center (CXC) is operated for NASA by the Smithsonian Astrophysical Observatory.
60 Garden Street, Cambridge, MA 02138 USA.

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