Language and Scripting in CIAO 4

What does it mean that Sherpa and ChIPS embed S-Lang and Python?

How do I “script” CIAO?

Doug Burke for the CXC

Language and Scripting in CIAO 4

S-Lang

`http://www.s-lang.org/` from CIAO 2.0

Python

`http://www.python.org/` from CIAO 4.0

`http://numpy.scipy.org/`

Tcsh

`http://www.tcsh.org/`

Bash

`http://www.gnu.org/software/bash/`

Also sh and bash set-up scripts available.

Language and Scripting in CIAO 4

Scripting tools

```
% source /soft/ciao-4.0/bin/ciao.csh
CIAO configuration is complete...
 CIAO version          :  CIAO 4.0.2 Friday, March 28, 2008
 Proposal Toolkit version :  Cycle 10 Thursday, ...
 bin dir               :  /soft/ciao-4.0/bin

% echo $PFILES
/Users/doug/cxcds_param4;/soft/ciao-
4.0/contrib/param:/soft/ciao-4.0/param
% paccess dmkeypar
/Users/doug/cxcds_param4/dmkeypar.par
% plist dmkeypar

Parameters for /home/dburke/cxcds_param4/dmkeypar.par
...
```

Scripting tools

```
% source /soft/ciao-4.0/bin/ciao.csh
CIAO configuration is complete...
 CIAO version             :  CIAO 4.0.2 Friday, March 28, 2008
 Proposal Toolkit version :  Cycle 10 Thursday, ...
 bin dir                  :  /soft/ciao-4.0/bin

% echo $PFILES
/Users/doug/cxcds_param4;/soft/ciao-
4.0/contrib/param:/soft/ciao-4.0/param
% paccess dmkeypar
/Users/doug/cxcds_param4/dmkeypar.par
% plist dmkeypar

Parameters for /home/dburke/cxcds_param4/dmkeypar.par
...
```

Why is this highlighted?

Scripting tools - location of parameter files

If you want to run tools at the same time to take advantage of your nice new multi-CPU machine – e.g. multiple `reproject_aspect` calls - then you have to be careful about the location of the parameter files, otherwise you can not guarantee what values are being used. In this case, it is safest to use *separate* directories for each script; for example

```
% set d = \
     “/soft/ciao-4.0/contrib/param:/soft/ciao-4.0/param”
% setenv PFILES “/data/pfiles1:$d”
% ./reduce_data_and_write_paper.csh 8003
% setenv PFILES “/data/pfiles2:$d”
% ./reduce_data_and_write_paper.csh 8004
```

Scripting tools - mode parameter of tools

Virtually all tools have a mode parameter that determines how the tool finds out what parameters to use; the values are the somewhat-cryptic values of `"a"`, `"h"`, `"l"`, `"q"`, or `"ql"` and are described in “`ahelp parameter`” (which is a good read to learn other tricks of the trade).

When running from a script, you will probably want to use `mode=h` to avoid the tool trying to ask you for values and getting annoyed when you don't answer.

Language and Scripting in CIAO 4

Scripting tools – spaces in filenames

The Data Model syntax ("`ahelp dmsyntax`") allows you to send wonderful file names like

```
  evt.fits[energy>300,sky=region(foo.fits)][cols time,sky]
```

in to programs. Unfortunately the presence of spaces [*] and characters like ">" can confuse scripts, so you may need to protect file names when you call tools (e.g. surround them with quotes).

[*] In particular, spaces can really complicate the handling of stacks ("`ahelp stack`").

Scripting tools – adding a parameter-file interface

You could use the command-line parameter interface tools – e.g. `pget`, `pquery`, `pline` – to make a shell script work well with parameter files, but I suggest you don't. See

```
$ASCDS_INSTALL/binexe/wavdetect
```

for an example of how painful this can be. Use the S-Lang parameter interface instead (“`ahelp paramio`”).

Sherpa and ChIPS interactive environments I

```
% sherpa

-----------------------------------------------------
Welcome to Sherpa: CXC's Modeling and Fitting Package
-----------------------------------------------------
Version: CIAO 4.0

sherpa-1> Python commands here

% sherpa
slsh version 0.8.2-0; S-Lang version: 2.1.3
Copyright (C) 2005-2007 John E. Davis <jed@jedsoft.org>
This is free software with ABSOLUTELY NO WARRANTY.

sherpa-1> S-Lang commands here
```

Language and Scripting in CIAO 4

Sherpa and ChIPS interactive environments II

How do I chose whether to use the S-Lang or Python versions? ChIPS and Sherpa uses Python by default; to change to S-Lang try

```
% setenv SHERPA_SCRIPT_LANG slang
% setenv CHIPS_SCRIPT_LANG slang

% grep chips.shell ~/.chips.rc
chips.shell             : slang
```

(no version for sherpa.shell)

```
% alias ssherpa
        '/usr/bin/env SHERPA_SCRIPT_LANG=slang sherpa'
```

Sherpa and ChIPS interactive environments II

How do I chose whether to use the S-Lang or Python versions? ChIPS and Sherpa uses Python by default; to change to S-Lang try

```
% setenv SHERPA_SCRIPT_LANG slang
% setenv CHIPS_SCRIPT_LAN

% grep chips.shell ~/.chi
chips.shell              :
```

(no version for sherpa.shell)

In CIAO 4.1, you will be able to say

```
% sherpa -l slang

% chips -l python
```

```
% alias ssherpa
        '/usr/bin/env SHERPA_SCRIPT_LANG=slang sherpa'
```

Language and Scripting in CIAO 4

ChIPS example

```
chips-1> make_figure(
... "bcg.txt[cols z,mk]",{"line.style","noline"});
chips-2> set_curve({"symbol.style","square","symbol.size",2.0});
chips-3> log_scale(X_AXIS);
chips-4> set_plot_ylabel("m_k");
chips-5> print_window("plot");
```

S-Lang

```
chips-1> make_figure(
      "bcg.txt[cols z,mk]",["line.style","noline"])
chips-2> set_curve(["symbol.style","square","symbol.size",2.0])
chips-3> log_scale(X_AXIS)
chips-4> set_plot_ylabel("m_k")
chips-5> print_window("plot")
```

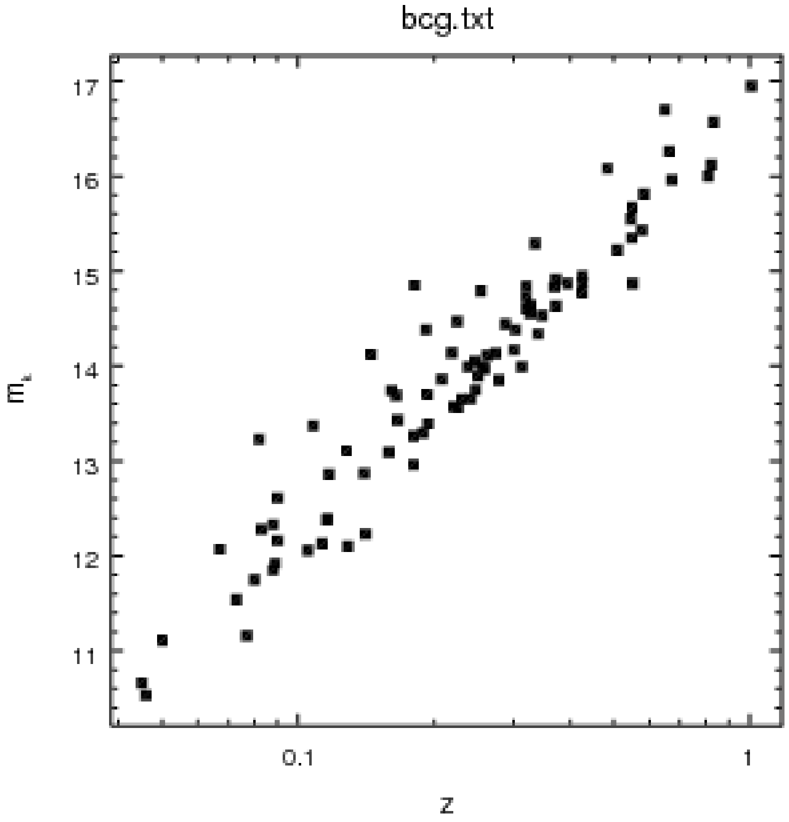
Python

Language and Scripting in CIAO 4

ChIPS example

```
chips-1> make_figure(
... "bcg.txt[cols z,mk]",{"line
chips-2> set_curve({"symbol.sty
chips-3> log_scale(X_AXIS);
chips-4> set_plot_ylabel("m_k")
chips-5> print_window("plot");
```

```
chips-1> make_figure(
          "bcg.txt[cols z,mk]",["l
chips-2> set_curve(["symbol.sty
chips-3> log_scale(X_AXIS)
chips-4> set_plot_ylabel("m_k")
chips-5> print_window("plot")
```



This example was chosen to highlight the similarities in the interfaces.

Language and Scripting in CIAO 4

An interactive environment

Here we use Crates to read in files (“`ahelp py.crates crates`” or “`ahelp sl.crates crates`”), manipulate it, then plot it:

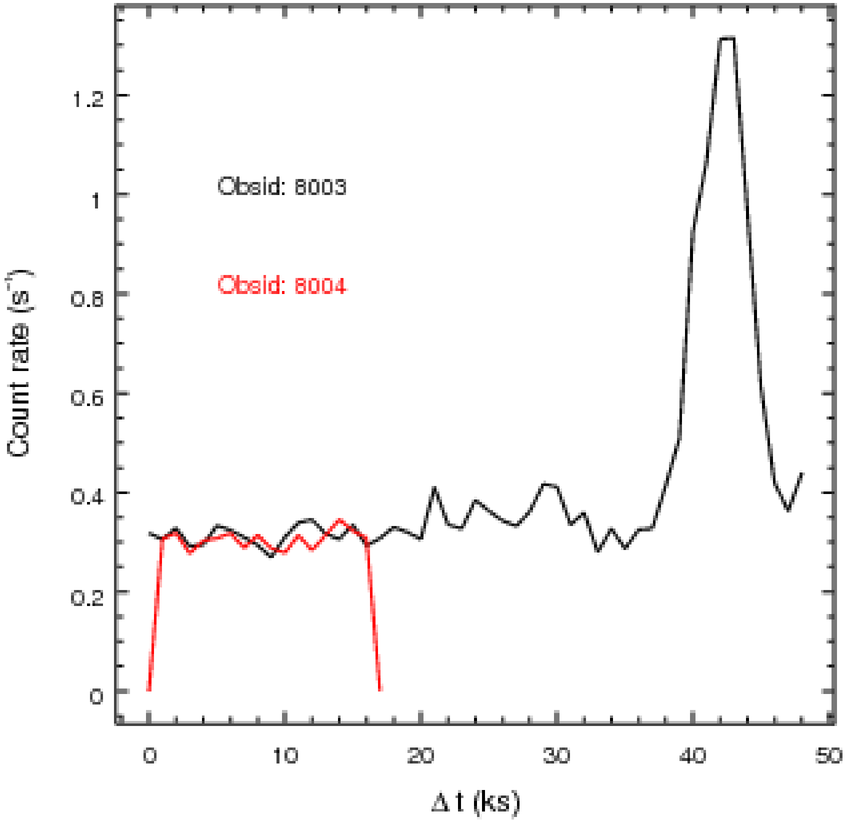
```
chips-1> a = read_file("8003/lcurve/ccd0.300-8000.d1000.lc")
chips-2> b = read_file("8004/lcurve/ccd1.300-8000.d1000.lc")
chips-3> xa = get_colvals(a,"time")
chips-4> ya = get_colvals(a,"count_rate")
chips-5> xb = get_colvals(b,"time")
chips-6> yb = get_colvals(b,"count_rate")
chips-7> xa = (xa - xa[0]) / 1000.0
chips-8> xb = (xb - xb[0]) / 1000.0
chips-9> add_curve(xa,ya,["symbol.style","none"])
chips-10> add_curve(xb,yb,
              ["symbol.style","none","line.color","red"])
...
```

Language and Scripting in CIAO 4

An interactive environment

Here we use Crates to read in f
sl.crates crates”), manipulat

```
chips-1> a = read_file("8003
chips-2> b = read_file("8004
chips-3> xa = get_colvals(a,
chips-4> ya = get_colvals(a,
chips-5> xb = get_colvals(b,
chips-6> yb = get_colvals(b,
chips-7> xa = (xa - xa[0]) /
chips-8> xb = (xb - xb[0]) /
chips-9> add_curve(xa,ya,["s
chips-10> add_curve(xb,yb,
              ["symbol.style'
...
```



help

Language and Scripting in CIAO 4

An interactive environment

You can run external programs or access ahelp:

```
chips-6> !ls
merged.fits ...
chips-7> !dmlist merged.fits counts
505950
chips-8> ahelp(“get_colvals”)       ;   [if S-Lang]
```

or load in other modules

```
sherpa-2> require(“stats”);              [S-Lang]

sherpa-2> from pylab import hist         [Python]
```

Writing Python or S-Lang scripts

The Sherpa and ChIPS environments load up useful modules (like ChIPS and Sherpa) but you have to do this yourself from a stand-alone script.

S-Lang:

```
#!/usr/bin/env slsh
require(“sherpa”);
require(“chips_hlui”);
require(“crates”);
```

Python:

```
#!/usr/bin/env python
from sherpa.astro.ui import *
from pychips.hlui import *
from pycrates import *
```

Language and Scripting in CIAO 4

Some missing topics…

What is the mysterious `~/.ipython-ciao/` directory?

What about the S-Lang modules like `paramio` and `pixlib`?

When is the CIAO contributed software going to update `xxx` or add a Python version of `yyy`?

Should I use S-Lang or Python?