

AH_{ELP} for CIAO 3.4

region

Context: modules

[Jump to: Description Example EXAMPLE OF USING THE REGION MODULE CHANGES IN CIAO 3.2](#)
[Bugs See Also](#)

Synopsis

The S–Lang interface to the CXC region library

Description

The region module is the interface between the S–Lang interpreter (see "ahelp slang") and the CXC region library (see "ahelp dmregions"). This document provides an overview of the features of the region module, and tips for using it efficiently in a S–Lang program. Detailed descriptions of each function are provided by individual ahelp pages.

The region 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("region");
```

Functions provided by the module

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

Return type	Function name	Arguments
Region_Type	regParse	String_Type
Short_Type	regInsideRegion	Region_Type, Double_Type, Double_Type
Array_Type	regInsideRegion	Region_Type, Array_Type, Array_Type
Double_Type	regArea	Region_Type
Double_Type	regArea	Region_Type, Double_Type, Double_Type, Double_Type, Double_Type, Double_Type
Double_Type, Double_Type, Double_Type, Double_Type	regExtent	Region_Type
String_Type	regRegionString	Region_Type
none	regPrintRegion	Region_Type

Using Region_Type variables

The region module defines a new variable type, `Region_Type`, which has a complex internal structure that cannot be printed using the S–lang `print()` command. To see the components of a `Region_Type` variable, use the `regRegionString()` or `regPrintRegion()` routines. `Region_Type` variables can be defined using the `regParse` command, which converts a CIAO region string into a `Region_Type` variable. Note that the region module does not have any intrinsic knowledge of WCS coordinates, so cannot be directly used to match pointed Chandra observations to specific regions. To convert between the various Chandra detector coordinates, read "ahelp pixlib".

Example

```
chips> require("region")
chips> circle_var = regParse("circle(10,10,4)")
chips> in_circle = regInsideRegion(circle_var, 10, 10)
chips> print(in_circle)
1
chips> in_circle = regInsideRegion(circle_var, 5, 5)
chips> print(in_circle)
0
chips> circle_area = regArea(circle_var)
chips> print(circle_area)
50.2655
chips> print(PI*4*4)
50.2655
chips> regPrintRegion(circle_var)
1 Circle(10.000000, 10.000000, 4.000000) (Pos: pixel, Size: pixel)
```

EXAMPLE OF USING THE REGION MODULE

The following routine can be used to calculate the total area in a CIAO region:

```
%
% Usage:
% result = calculate_region_area( region_string );
%
require("region");
define calculate_region_area ( region_string ) {

    variable reg = regParse( region_string );
    if (reg == NULL) {
        vmessage("Error: %s not a legal CIAO region.",region_string);
        return;
    }
    variable area = regArea(reg);
    vmessage("Area inside region = %e",area);
} % calculate_region_area( region_string )
```

With the above function, one can calculate the enclosed area inside a CIAO region string. Note that the region module does not have any built–in knowledge of the Chandra pointing or the detector field of view; therefore, the area of a region like "field()" is infinite (or `DBL_MAX`, as shown below):.

```
sherpa> evalfile("calculate_region_area.sl");
1
```

```
sherpa> calculate_region_area("circle(4,4,10)")
Area inside region = 3.141593e+02
sherpa> calculate_region_area("fred")
Error: Could not parse region string
Error: fred not a legal CIAO region.
sherpa> calculate_region_area("field()")
Area inside region = 1.797693e+308
```

CHANGES IN CIAO 3.2

The module can now be loaded by using the

```
require("region");
```

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

regExtent()

The regExtent() routine, which calculates the bounding box of a region, has been added to the module. See "ahelp regextent" and "ahelp regarea" for more information and examples of its use.

regRegionString()

The regRegionString() routine has been added to the module.

Bugs

See the [bugs page for the region library](#) on the CIAO website for an up-to-date listing of known bugs.

See Also

region

[regarea](#), [regextent](#), [reginsideregion](#), [regparse](#), [regprintregion](#), [regregionstring](#)

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/region.html>
Last modified: December 2006

Ahelp: region – CIAO 3.4