



---

*AHELP for CIAO 3.4*

## get\_groups

Context: [sherpa](#)

*Jump to:* [Description](#) [Examples](#) [Bugs](#) [See Also](#)

---

### Synopsis

Module functions for retrieving a grouping or quality array from source and background files.

### Syntax

```
Integer_Type get_groups([Integer_Type])
Integer_Type get_bgroups([Integer_Type])
Integer_Type get_quality([Integer_Type])
Integer_Type get_bquality([Integer_Type])

Success/Error Return Values: 1/0

Arguments:

(1) Dataset number (default 1).
```

### Description

The `get_(b)groups` and `get_(b)quality` functions allow the user to retrieve grouping and quality information for source and background datasets.

If the file is not grouped or quality is not set, NULL is returned. If the files is grouped or quality defined, an array is returned.

The retrieved array is a `Integer_Type` array of the same length as the input (grouped) dataset. A grouping array element set to `-1` marks the beginning of a group, while array element set to `1` marks members of that group, so the corresponding bins are treated as one during fitting.

A quality array contains the quality flags for each group: `0` for good (grouped) data; `5` for data labeled as bad by the user (within a tab), and `2` for data labeled as questionable by `dmgroup` (incomplete groups, etc.).

The grouping and quality definitions are based on OGIP standard.

See the related Sherpa commands `GROUP` and `QUALITY` for more information.

## Example 1

Retrieve and apply a grouping scheme from the other data set to an ungrouped data set.

```

sherpa> DATA spec.pha
sherpa> show
.....
-----
Input data files:
-----

Data 1: spec.pha pha.
Total Size: 1024 bins (or pixels)
Dimensions: 1
Total counts (or values): 2231

.....
sherpa> DATA 2 spec_grp.pha
sherpa> show
.....
-----
Input data files:
-----

Data 1: spec.pha pha.
Total Size: 1024 bins (or pixels)
Dimensions: 1
Total counts (or values): 2231

.....

Data 2: spec_grp.pha pha.
Total Size: 131 bins (or pixels)
Dimensions: 1
Total counts (or values): 2231
.....
sherpa> g=get_groups(2)
sherpa> print(g)
1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
1
-1
.....
sherpa> set_groups(1,g)
WARNING: any applied filters are being deleted!

```

```

1
sherpa> show

Data 1: spec.pi pha.
Total Size: 131 bins (or pixels)
Dimensions: 1
Total counts (or values): 2231
.....

```

In this example, ungrouped and group data are read into Sherpa, and then group information is retrieved using `get_groups` from the grouped dataset. A new array `g` is defined whose elements are defined by the grouping of the dataset 2. This grouping scheme is then apply to the ungrouped data set with `set_groups`.

## Example 2

Retrieve and apply a grouping scheme from the source data set to ungrouped background data.

```

sherpa> DATA spec.pha
sherpa> show
....
-----
Input data files:
-----

Data 1: spec.pha pha.
Total Size: 1024 bins (or pixels)
Dimensions: 1
Total counts (or values): 2231

Background 1: bg.pha pha.
  Total Size: 1024 bins (or pixels)
  Dimensions: 1
  Total counts (or values): 662

.....

sherpa> g=get_groups(1)
sherpa> print(g)
1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
1
-1
.....
sherpa> set_bgroups(1,g)
WARNING: any applied filters are being deleted!

```

```
1
sherpa> show

Data 1: spec.pi pha.
Total Size: 131 bins (or pixels)
Dimensions: 1
Total counts (or values): 2231

Background 1: bg.pha pha.
Total Size: 131 bins (or pixels)
Dimensions: 1
Total counts (or values): 662
.....
```

In this example, group data and ungrouped background data are read into Sherpa, and then group information is retrieved using `get_groups` from the grouped dataset. A new array `g` is defined whose elements are defined by the grouping of the source data set. This grouping scheme is then apply to the ungrouped background data set with `set_bgroups`.

## Bugs

See the [Sherpa bug pages](#) online for an up-to-date listing of known bugs.

## See Also

*sherpa*

[analysis](#), [ignore](#), [notice](#), [set\\_filter](#), [set\\_groups](#), [set\\_ignore](#), [set\\_ignore2d](#), [set\\_ignore\\_all](#), [set\\_ignore\\_bad](#), [set\\_notice](#), [set\\_notice2d](#), [set\\_notice\\_all](#)

---

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/get\\_groups.html](http://cxc.harvard.edu/ciao3.4/get_groups.html)  
Last modified: December 2006