
Subject: "% Unable to allocate memory" when using histogram.
Posted by [Paul Van Delst\[1\]](#) on Tue, 20 Oct 2015 15:19:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

I have a utility that plots histograms of satellite data. I mention this because satellites==lots-of-data. (Although in this case, not so much)

After reading in all the data I see the following:

```
IDL> help, boxcar_obs, srf_obs
BOXCAR_OBS   STRUCT  = -> <Anonymous> Array[10600]
SRF_OBS      STRUCT  = -> <Anonymous> Array[10600]
```

Each element of the structure looks like:

```
IDL> help, BOXCAR_OBS[0]
** Structure <5424ac18>, 18 tags, length=1936, data length=1936, refs=2:
  OBS          FLOAT   Array[22]
  DEPAR_BC     FLOAT   Array[22]
  DEPAR_NBC    FLOAT   Array[22]
  ...etc...
  BIAS_SCANANGLE_TERMS
                        FLOAT   Array[22, 5]
  BIAS_SST_TERM  FLOAT   Array[22]
```

So, 18 tags of 22 element arrays of floats, except for the [22,5] array.

I calculate this to be
 $4 * 22 * 22 = 1936$ bytes
just as reported above.

So, there are 10600 of these, and two sets of data. Thus

$$1936 * 10600 * 2 = 41\text{MB}$$

No worries, right? The memory report looks similar:

```
IDL> memory(/structure)
{
  CURRENT: 47076590,
  NUM_ALLOC: 14800356,
  NUM_FREE: 14798590,
  HIGHWATER: 47076645
}
```

with 47MB reported.

For testing, I then did the following:

```
n = 100
boxcar_obs = boxcar_obs[0:n]
srf_obs = srf_obs[0:n]
memory(/structure)
```

and I got:

```
{
  CURRENT: 3904579,
  NUM_ALLOC: 16682008,
  NUM_FREE: 16680242,
  HIGHWATER: 47271996
}
```

Again, no worries. Makes sense, yeah?

Then, as I loop over each of the 22 channels I call the histogramming utility, ala:

```
for i = 0, nchan-1 do begin

  gsi_radstat_histogram, $
    boxcar_obs, $
    obsdata2 = srf_obs, $
    channel_index=i

endfor
```

I added memory output right before the histogram function is called. And I get:

channel 1

** Structure IDL_MEMORY, 4 tags, length=16, data length=16:

CURRENT	LONG	3910048
NUM_ALLOC	LONG	21910192
NUM_FREE	LONG	21908421
HIGHWATER	LONG	3929559

channel 2

** Structure IDL_MEMORY, 4 tags, length=16, data length=16:

CURRENT	LONG	7176461
NUM_ALLOC	LONG	22116633
NUM_FREE	LONG	22110660
HIGHWATER	LONG	7185333

channel 3

** Structure IDL_MEMORY, 4 tags, length=16, data length=16:

CURRENT	LONG	7176461
NUM_ALLOC	LONG	22296048
NUM_FREE	LONG	22290075
HIGHWATER	LONG	7185333

% Unable to allocate memory: to make array.

Cannot allocate memory

% Execution halted at: GSI_RADSTAT_HISTOGRAM 93

The offending line in question is:

```
pdf = HISTOGRAM( tb_data, $  
                BINSIZE = _binsize, $  
                LOCATIONS = xbin )
```

The error occurs at the third channel even if I DO NOT subset the data, keeping all 10600 elements e.g.:

channel 1

** Structure IDL_MEMORY, 4 tags, length=16, data length=16:

CURRENT	LONG	47165566
NUM_ALLOC	LONG	24028618
NUM_FREE	LONG	24026847
HIGHWATER	LONG	48098350

channel 2

** Structure IDL_MEMORY, 4 tags, length=16, data length=16:

CURRENT	LONG	50431979
NUM_ALLOC	LONG	24341060
NUM_FREE	LONG	24335087
HIGHWATER	LONG	51364763

channel 3

** Structure IDL_MEMORY, 4 tags, length=16, data length=16:

CURRENT	LONG	50431979
NUM_ALLOC	LONG	24658491
NUM_FREE	LONG	24652518
HIGHWATER	LONG	51364763

It would appear (to me at least) that the physical sizes of my arrays in question are not at issue here - I get the unable-to-allocate error even if I am using a fraction of the total.

Does anyone have any idea what could be causing this error? And how to fix it?

Thanks,

paulv

p.s. I have a typical linux redhat system, 4GB memory, RHEL6

```
IDL> !version
```

```
{
  ARCH: "x86_64",
  OS: "linux",
  OS_FAMILY: "unix",
  OS_NAME: "linux",
  RELEASE: "8.3",
  BUILD_DATE: "Nov 15 2013",
  MEMORY_BITS: 64,
  FILE_OFFSET_BITS: 64
}
```

p.p.s. Potential red herring: I was having licensing issues the day before yesterday that required me to restart (to get Imgrd working properly). Unfortunately I didn't run this code that fails until today.

p.p.p.s. I have a test program for this particular histogram utility into which I put memory output as it generates six histograms:

```
IDL> test_gsi_radstat
```

```
% Compiled module: TEST_GSI_RADSTAT.
```

```
% Compiled module: READ_DIAGS.
```

```
% Compiled module: GSI_RADSTAT_HISTOGRAM.
```

```
** Structure IDL_MEMORY, 4 tags, length=16, data length=16:
```

```
  CURRENT      LONG      412374977
  NUM_ALLOC     LONG      1308019
  NUM_FREE      LONG      1307012
  HIGHWATER     LONG      421668193
```

```
** Structure IDL_MEMORY, 4 tags, length=16, data length=16:
```

```
  CURRENT      LONG      418893819
  NUM_ALLOC     LONG      1436201
  NUM_FREE      LONG      1417730
  HIGHWATER     LONG      428187035
```

```
** Structure IDL_MEMORY, 4 tags, length=16, data length=16:
```

```
  CURRENT      LONG      420167967
  NUM_ALLOC     LONG      1538103
  NUM_FREE      LONG      1512960
  HIGHWATER     LONG      429461183
```

```
** Structure IDL_MEMORY, 4 tags, length=16, data length=16:
```

```
  CURRENT      LONG      422355746
  NUM_ALLOC     LONG      1673174
```

NUM_FREE	LONG	1639501
HIGHWATER	LONG	431648962
** Structure IDL_MEMORY, 4 tags, length=16, data length=16:		
CURRENT	LONG	423641670
NUM_ALLOC	LONG	1775399
NUM_FREE	LONG	1735101
HIGHWATER	LONG	432934886
** Structure IDL_MEMORY, 4 tags, length=16, data length=16:		
CURRENT	LONG	424951845
NUM_ALLOC	LONG	1877609
NUM_FREE	LONG	1830749
HIGHWATER	LONG	434245061

Note the much higher memory usage but lower NUM_ALLOC and NUM_FREE.
