Subject: Re: Misc. Bugs & Problems
Posted by steinhh on Thu, 25 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

In article <eK6N4yBY#GA.322@ntawwabp.compuserve.com>
"Steve Scheele" <sscheele@scitor.com> writes:

> I am running IDL V5.2 on WinNT and have encountered the following bugs and
> problems. All have been discussed with RSI technical support.
>
> ;**********
> Problem: Sort slows down considerably when sorting integer arrays
> containing many identical values.
>
> Workaround: Add small-random values to the array before sorting -
> improves sorting by 40X

Note that this seems to be platform dependent. I think IDL is using
whatever builtin sorting routine exists on the specific platform. On
Digital Unix:

```
IDL> .run
  pro test,n
  i=fix(randomu(seed,100000)*n)
  t=systime(1)
  ix=sort(i)
  print,systime(1)-t
  end

IDL> test,10     ;; Lots of identical values
      0.12988293
IDL> test,100
      0.31445301
IDL> test,1000
      0.46289098
IDL> test,10000
      0.65039003
IDL> test,100000 ;; Fewest identical values.
      0.77246106
```

So, adding a random value to make each entry unique could increase
the execution time here by an order of 10 (ok, I'm an astronomer!).

I seem to remember that we found some pathological case for which the
Unix sorting routine (qsort?) was very slow...

Ah, yes, this is it:

```
IDL> i=randomu(seed,10000)
IDL> t=systime(1) & ix=sort(i) & print,systime(1)-t
     0.059569955
IDL> i=[findgen(5000L),findgen(5000L)]
IDL> t=systime(1) & ix=sort(i) & print,systime(1)-t
       9.2021489
```

Ouch! Randomizing the order of the numbers before sorting would give a speed increase on the order of 100!!

There will always be pathological cases for a given algorithm, though. It would be nice, however, if the pathological cases for IDL's SORT() was *known*, though, independently of the platform, so one could avoid them when the prior structure of the dataset is known.

Maybe the best thing would be to add keywords to SORT() that advises about any known structure... Like /DUPLICATES to warn about many identical values, /IDENTICAL_SEQUENCES to warn about long, identical, sorted subsequences, etc etc...?

> Bug: Passing a UINT array to REBIN crashes IDL

Wheee !

> Bug: Resizing a draw widget, flips vertical sliders up side down.
>
> Workaround: Pass an initial value to the slider - this workaround is
> apparently machine/OS dependent. It didn't work for me.

Could somebody give a tiny example of this - I'm not exactly clear on what this means.

Regards,

Stein Vidar

Subject: Re: Misc. Bugs & Problems
Posted by menakkis on Thu, 25 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

"Steve Scheele" <sscheele@scitor.com> wrote:
> ;**********
> Bug: Resizing a draw widget, flips vertical sliders up side down.
>
> Workaround: Pass an initial value to the slider - this workaround is
> apparently machine/OS dependent. It didn't work for me.

I think I have a genuine workaround for NT. Using IDL 5.2 / WinNT, with a widget that contains a menu, draw widget and vertical slider the full height of the draw widget... I initially create the slider with: sl=widget_slider(b0,min=0L,max=10000L,val=10000L,/suppress,u val=2,/vert) When handling the main widget's resize call, I find I have to do: widget_control,sl,set_slider_max=0L,set_slider_min=10000L The astute reader will notice that the min and max are now the reverse of what was used in the creation call, and that seems to balance out whatever happened to the poor thing during the resize. (Well, on NT at least.) I think it keeps on working too (i.e., for any further resizes).

Peter Mason