Subject: Misc. Bugs & Problems
Posted by Steve Scheele on Wed, 24 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

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

;**********
Bug: Passing a UINT array to REBIN crashes IDL

Workaround: Don't do that

;**********
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.

;**********
Problem: The IDL Code printer font size is proportional to, larger than, but not the same size as display font. There is no independent control over the printer font. Making the problem worse is fact that IDL prints line numbers with the code. I use a 10pt display size which results in an 11.5pt print size which runs the printed code off the right margin. As a result the IDL code printer is worthless to me.

Workaround: Use another program to print the code - large pain!

;**********
Problem: In spite of RSI documentation to the contrary, IDL has no true global variables.

Workaround: User defined system variables - RSI really doesn't like this workaround

;**********
Bug: RANDOMN has a bug which can cause HISTOGRAM (with a variable BINSIZE) to fail. When RANDOMN and HISTOGRAM are both in a long For loop, a "corrupt array descriptor" error message is eventually displayed. This bug is related to calling RANDOMN with an uninitialized or variable Seed.

Workaround: Pass BINSIZE as a string to HISTOGRAM

;********************
Problem: TrueType display fonts look lousy

Workaround: None

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

I'll give it a try.

<menakkis@my-dejanews.com> wrote in message
news:7b2mg7ua1@nnrp1.dejanews.com...
> "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
>
> ----------== Posted via Deja News, The Discussion Network ==----------
> http://www.dejanews.com/ Search, Read, Discuss, or Start Your Own

---

Subject: Re: Misc. Bugs & Problems
Posted by Steve Scheele on Thu, 25 Feb 1999 08:00:00 GMT

< Problem: Sort slows down considerably when sorting integer arrays
> containing many identical values. (on NT)

> Note that this seems to be platform dependent.

Yes - It is also my understanding that Sort uses the sort algorithm that is native to the particular OS. I had thought that Sort used quick sort. However, quick sort slows down considerably when sorting arrays already in (mostly) sorted order. My testing failed to show this particular problem.

< Bug: Resizing a draw widget, flips vertical sliders up side down.
< Workaround: Pass an initial value to the slider

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

The test case I have been using is;

```
;****************************************************** ******
 Pro TestSlider

   TLB = Widget_Base(/Row, XPad=10, /Map, XSize=200, YSize=200)

   Slider = Widget_Slider(TLB, Maximum=100, Minimum=0, /Vertical)

   Draw = Widget_Draw(TLB)

   ;following causes the slider to flip 180 deg., putting the min value at
top and the max value at the bottom
   Widget_Control, Draw, Draw_XSize=75

   XManager, 'TLB, TLB

 End
```

RSI's workaround is to build the slider with an explicit value, i.e.

```
    Slider = Widget_Slider(TLB, Maximum=100, Minimum=o, /Vertical,
Value=50)
```

This didn't work for me. Apparently both the problem and workaround are IDL version and OS (and OS version) specific. My test code worked proberly using IDL V5.1 on both NT and SUN/Solaris.

Subject: Re: Misc. Bugs & Problems
Posted by steinhh on Fri, 26 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Steve Scheele wrote:

> The test case I have been using is;
[...]
> ;following causes the slider to flip 180 deg., putting the min value at
> ;top and the max value at the bottom
> Widget_Control, Draw, Draw_XSize=75
[...]

Doesn't happen on my machine... I always get minimum value at
bottom, on { alpha OSF unix 5.2 Oct 30 1998} with Mwm...

Regards,

Stein Vidar

Subject: Re: Misc. Bugs & Problems
Posted by David Kastrup on Fri, 26 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

"Steve Scheele" <sscheele@scitor.com> writes:

> < Problem: Sort slows down considerably when sorting integer arrays
>> containing many identical values. (on NT)
>
>> Note that this seems to be platform dependent.
>
> Yes - It is also my understanding that Sort uses the sort algorithm
> that is native to the particular OS. I had thought that Sort used
> quick sort. However, quick sort slows down considerably when
> sorting arrays already in (mostly) sorted order. My testing failed
> to show this particular problem.

Only naive quicksort slows down on sorted elements. The usual variant
is the median of three quicksort picking the pivot from a median of
first, last and middle element of a range. The most inefficient
configuration for this sort is pretty hard to come up by design, but
is also O(n^2).

When you are using the GNU C library, qsort really reverts to a
mergesort method unless memory is scarce.

--
David Kastrup Phone: +49-234-700-5570
Email: dak@neuroinformatik.ruhr-uni-bochum.de Fax: +49-234-709-4209
Institut fï¿½r Neuroinformatik, Universitï¿½tsstr. 150, 44780 Bochum, Germany