Subject: Plea for IDL 2000 (was: a plea for more reliable mathematical routines)
Posted by m218003 on Thu, 16 Sep 1999 07:00:00 GMT

View Forum Message <> Reply to Message

In article <MPG.124804e8864d03df9898ec@news.frii.com>,
>>
>> I definitely disagree. It is inferior to Java, Python, C/C++
>
> I mean, honestly, that fact that IDL is still selling as well
> as it does is not a testament to what a great language it is.
> It is a testament to how hard it is to write something like
> it that can beat it in the marketplace.
> Somebody has to be looking at the ol' man and thinking
> "I can do better than that." Perhaps that somebody is
> you, Arno. [...][/color]

Sound to me as if Dave Stern should take this as a wakeup call and
lock himself into the garage for a while ;-) Seriously: RSI has the
source code, so for them it would be a head start to come up with a
more modern version of IDL -- let's just name it IDL 2000 (although
it might be a little late for this). Who really needs more features in
IDL as it is? Why not consolidate all forces and give the old lady
(why "man" David?) a rejuvenalization? Start from scratch and rebuild
the language, but leave the general syntax (arguments and keywords,
comma seperation) untouched, so it would be somewhat less hard to
convert old programs (of course even better would be some kind of
conversion tool which would be helpful enough if it could convert
90% of the code and mark the remaining 10%).

As a start, here a short list of fundamentals that I would like to
see changed:

* Data types: The default integer should be 32 bit (if not even 64),
  so you would get rid of many problems with loops and array indices.
  For reading of binary files, a 2 byte integer should be kept as
  SHORT.

* Mathematical routines: well, see the thread. But definitively, they
  should all operate on DOUBLE as default and -- if at all -- allow a
  /SINGLE keyword in case one runs out of memory with DOUBLE arrays.

* PLOT and the likes: should also operate with DOUBLE values. There
  should be a solid 2D object oriented model together with a successor
  of the direct graphics routines (which are still useful to create
  real quick overview plots and maybe even for publication quality
  plots exactly because of the lack of interactivity!).
  All the keywords should be cleaned, e.g. it should be possible
  to call PLOT with an /OVERPLOT keyword (i.e. the same as OPLOT)

and you should definitively not need an /ADVANCE keyword for MAP_SET. Also (as I remarked about a year ago) it should be more intuitive e.g. to turn of axis labels (would you have ever thought of TICKFORMAT='(A1)' on your own?). AND: please get rid of these annoying character units (e.g. MARGIN)!

* SMP (symmetric multiprocessing) is becoming more and more of a standard, so IDL should support it.

* Foreign language support in the vector fonts would also be on my wish list (but please don't go as far as Windows where you always have to access the system's control panel to change from ',' as decimal notation to '.' if you exchange data with colleagues in an Excel spreadsheet). It would just be nice if one could e.g. write "o for an o-umlaut (perhaps it would be great if the Tex2IDL interface would become some integral part of IDL. TeX is extremely powerful in formatting equations etc., and it is much more intuitive to write 'A_3' or 'A_{3,2}' instead of 'A!L3!N' or 'A!L3,2!N'. Also, TeX is fairly wide spread so people would not have to adapt to yet another formatting syntax.

* The image routines should be standardized and not consist of a mixture between procedures and functions as it is now.

* In order to achieve more consistency between the object oriented graphics and direct graphics, one should perhaps think of adapting the GNUPLOT approach: there you call a bunch of SET commands to specify the plot style, and then the PLOT command becomes much shorter. In principle one could map the object structure with its view, plotwindow, axis, etc. onto a hierachical structure which could be used in the same way by the direct graphics routines (probably direct graphics would in the end be only another interface for what in fact is object graphics?).

Once again: I really think IDL as we know it has reached its maturity a while ago, and now it's time to think of the future! Of course RSI still needs to support the current IDL version and its users, but I really think they should give their development team a new direction.

Have a good day,

Martin

--
[[[[[[[[[
[[Martin Schultz   Max-Planck-Institut fuer Meteorologie   [[
[[Bundesstr. 55, 20146 Hamburg   [[

[[phone: +49 40 41173-308 [[
[[fax: +49 40 441787 [[
[[martin.schultz@dkrz.de [[
[[[[[[[[[[[[[[[