
Subject: Giant arrays!

Posted by [Ben Krasnow](#) on Fri, 27 Jun 1997 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi all! I am new to PV-WAVE, and I have a 'can it do this' question: I made a simple program that loads a three dimensional(256x256x30 images) image array, selects one image, and then uses assoc to 'copy' one of the images to a two dimensional array. Next I want to transfer that into a one-dimensional array so I can use sort on it. the problem is the one-dimensional array has to be 65536 elements, to hold the single 256x256 image's data. PV-WAVE bombs, and says something to the tune of 'too many elements'. Is there another way to do this, without breaking the array up(I have to use sort on it)? Thanks

Subject: Re: Giant arrays!

Posted by [David Ritscher](#) on Fri, 11 Jul 1997 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Mary Jo Brodzik writes:

>> P.S. Anybody else want to join in a campaign for a system variable that
>> switches the default literal integer to long rather than short? I've
>> been bit too many times...

> Yes, now that would be useful, I've been bitten a number of times,
> myself. And although I try to warn new IDL users, this problem
> continues to bite them, too!

Do you mean that

`i = 2`

should produce a something other than a 2-byte integer?

I think we should discuss this problem after the year 2000 :-)

IDL 5.0 has finally fixed the 'enhanced feature' of not being able to differentiate between an array `a(3)` and a function call `my_function(3)`. I think it is time for a language reform, where a system variable tells the interpreter when to interpret code as vintage code, so, for example,

`!LANGUAGE_LEVEL = 1`

at the beginning of a routine (i.e., one I wrote years ago) would tell the interpreter to default to 2-byte integers, arrays with parenthesis for delimiters, etc. With such a mechanism in place, then RSI and VNI would be free to update the basic language structure while leaving our old code in a functional state.

What would other good additions entail? A syntax that follows C++ more closely would be a good start (concerning use of '(', '[', '{', etc.) Other suggestions??

Tschuess,

David Ritscher

--

David Ritscher

Zentralinstitut fuer Biomedizinische Technik Tel: ++49 (731) 502 5313

Albert-Einstein-Allee 47

Fax: ++49 (731) 502 5315

Universitaet Ulm

Internet:

D-89069 ULM

david.ritscher@zibmt.uni-ulm.de

Germany
