David Fanning <davidf@dfanning.com> wrote:
> William (wmc@bas.ac.uk) writes:
>> If so, could this be extended to allow *any* structures to be put into
>> arrays? Is there any particular reason why all array elements have to
>> be the same type of structure?

> This is the idea behind the changes in IDL 5.4. As long
> as the array elements "match" in the sense of having the
> same amount of storage allocated to them, structures
> can be put into arrays, etc.

Ah, no, thats not enough. I want, say, structures which represent a data
agglomeration (I'm trying not to say object) with loads of header fields
the same, but a few fields (maybe just the one "data" field) different.
So the storage is different. I know I could do this by putting a pointer
into the structure instead, but... I can't see why IDL shouldn't do this itself.
I guess I'm assuming that, when IDL stores an array of strucutures, it doesn't
store the structures consecutively anyway - just pointers to the structures.
In which case, it shouldn't matter what the structure types are. I think.

-W.

ps - I'll hide this here so as to not start a perl/idl war: I use both a
lot and I like the syntax of both. But... occaisionally the clunkiness
of IDL does irritate a little. Like, not allowing null arrays.

--
William M Connolley | wmc@bas.ac.uk | http://www.nerc-bas.ac.uk/icd/wmc/
Climate Modeller, British Antarctic Survey | Disclaimer: I speak for myself