Subject: bytarr type conversion/structures
Posted by Jacobus Koster on Fri, 25 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Here's one for the pro's (and DF :-), at least, I think, but I'm no (big) expert.
Suppose my image files have a standard format with a standard 2048 byte header.
This header consists of 10 bytes of extraneous information followed by a descriptor part and a data part (don't worry, I never took a data structure class so all my terminology is hopelessly inaccurate)
The descriptor part consists of about 100 descriptor fields, each of which is a record/structure consisting of 3 short ints : key.type, key.length, and key.offset.
key.type is a code indicating the datafield is string, shortint, char, etc.
key.length tells you the length of the data field in bytes
key.offset is the offset of the start of the data field, also in bytes.
This is all fairly standard sofar, I would think.
Now :
I would like to read these headers as byte arrays of 2048 bytes, and then forget forever about the file I got them from. From this byte array, I want to read the 100 descriptor structures into a 100-element structure array, with the structure elements described by : {type:0,length:0,offset:0}. And then, I would like to access the actual data itself, of course.
Is it possible in IDL to do this kind of type conversion, WITHOUT first writing the byte array out again into a dummy file and using an - albeit very beautiful - ASSOC variable or something like that ?
All I know is : it should involve pointers (if only because the type of the actual data varies). In addition to this, I would like to avoid FOR-loops, and explicitly calculating integers as 256*byte1 + byte2 strikes me as kinda crude.
Ponder this over your weekend, or answer right away if you like. TIA.

Sjaak Koster