

---

Subject: Re: matching 'C' Data Types?

Posted by [thompson](#) on Fri, 20 Oct 1995 07:00:00 GMT

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

---

Peter Sharer <[sharer@argus.arc.nasa.gov](mailto:sharer@argus.arc.nasa.gov)> writes:

> I am writing an IDL application which reads data from an unformatted  
> binary data file written by a 'C' program. Everything runs on an SGI  
> Indigo 2. I seem to be having a problem matching a few of the C data  
> types, specifically unsigned long and unsigned short. Which IDL types  
> map to those C types? I would greatly appreciate any ideas anyone  
> has.

IDL doesn't have an unsigned data type. Unsigned integers are stored exactly the same as signed integers. The only difference is the way the topmost bit is interpreted. If that bit is zero, i.e. there are no numbers above  $2^{15}-1=32767$ , then the distinction between signed and unsigned short integers can be ignored.

We read unsigned short integers into ordinary short integers, and then call the following to convert them into long integers so that numbers above 32767 are not misinterpreted as negative numbers.

```
VALUE = LONG(VALUE) AND 'FFFF'X
```

For unsigned long integers, I'm not sure what to suggest. If the numbers never rise above  $2^{31}-1=2147483647$  you can ignore the distinction between signed and unsigned long integers. Otherwise, if you want to preserve all 32 bits of precision in the number, you can read it in as a long integer, and then convert the number to double precision via the following:

```
VALUE = BYTE(VALUE, 4, N_ELEMENTS(VALUE)/4)  
C = 256.D0  
VALUE = VALUE(0,*)*C^3 + VALUE(1,*)*C^2 + VALUE(2,*)*C + VALUE(3,*)
```

Depending on how your machine works, you may need to reverse the order of the bytes in the above.

Bill Thompson

---

---

Subject: Re: matching 'C' Data Types?

Posted by [joseph.b.gurman](#) on Sun, 22 Oct 1995 07:00:00 GMT

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

---

In article <[4692eu\\$1to@hammer.msfc.nasa.gov](mailto:4692eu$1to@hammer.msfc.nasa.gov)>, [mallozzi@ssl.msfc.nasa.gov](mailto:mallozzi@ssl.msfc.nasa.gov) wrote:

>> I seem to be having a problem matching a few of the C data

>> types, specifically unsigned long and unsigned short.

>

> C datatype sizes:

>

> short 2 bytes

> int 4

> float 4

> long 4

> double 8

Note that this is platform-dependent; e.g., DEC Alphas (and I assume 64-bit MIPS

machines as well) would have 8-byte longs and 16-byte(?) doubles.

>

> IDL datatype sizes:

>

> byte 1 byte

> integer 2

> float 4

> long 4

> double 8

>

Note also that (1) the byte data type is the only unsigned data type in IDL, and (2) that IDL preserves the above across platforms (whether this is good or bad probably depends on your application). Frankly, I hope it changes as SGI, HP, and others finally move to 64-bit OS's.

> Pretty sure these are correct. I was doing exactly what you are doing

> with c and IDL, but it was quite awhile ago...

Joe Gurman

--

| No time to be witty; sorry |