Subject: Re: UNsigned Integer Data
Posted by Struan Gray on Thu, 06 Feb 1997 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning, davidf@dfanning.com writes:

> ```
> array = LONG(array) AND 'FFFF'x
> ```
>
> Now you have an array of LONG integers, but they
> have the correct unsigned values. There is no way to get
> around the requirement for LONG integers unless your data
> is always between 0 and 2^31-1 or 2147483647

If you have unsigned, 2-byte integers that go:

```
0000 0000 0000 0000   :   0
1000 0000 0000 0000   :   32768
1111 1111 1111 1111   :   65535
```

All you really need to do to get signed integers is
twiddle the top bit:

```
1000 0000 0000 0000   :   2s-comp -32768
0000 0000 0000 0000   :   2s-comp  0
0111 1111 1111 1111   :   2s-comp  32767
```

Any routines that calculate a real-world value (ie to label
and axis on a graph) need to have an offset part that accounts
for the fact that 'zero' has moved, but for a lot of datasets
that's a more convenient way of handling things anyway.

Thus you can avoid the memory and time penalty of converting
to longs with something like this:

```
array = temporary(array)  xor fix(-32768)
```

It definitely works faster on our macs, and since some of
our data files are pushing the memory limits anyway it saves
a lot of disk-swapping for the big ones.

Struan