Subject: IDL's handling of LOGICAL quantities (WHERE)
Posted by James Tappin on Tue, 12 Oct 1999 07:00:00 GMT
View Forum Message <> Reply to Message

\begin{rant}
I've finally decided to have a public moan about one of the weaknesses of IDL's handling of logical operations: to boot -- that the WHERE function follows a C-like interpretation while most other things are Fortran-like.

for example suppose we have an array (m) some of whose values are NaN then the (inefficient) loop:
for j=0, n_elements(m) do if not finite(m(j)) then m(j)=0
will set all non-finite elements of m to 0.
However:
m(where(not finite(m))) = 0
will zero out the whole array since where sees (not 1) as a Yes.
[The correct solution is of course:
m(where(finite(m) ne 1)) = 0
]

Or a simpler example:
```
IDL> a = [0, 1, 0, 1]
IDL> print, where(a eq 0)
           0           2
IDL> print, where(not (a ne 0))
           0           1           2           3
```

I guess the proper answer isto have aproper logical or boolean type and functions like FINITE and logical operations should return it, and of course WHERE should accept it.

\end{rant}

--
+----------------------+-----------------------------+-- -------+
James Tappin	School of Physics & Astronomy	O__
sjt@star.sr.bham.ac.uk	University of Birmingham	-- \/`
Ph: 0121-414-6462. Fax: 0121-414-3722		
+--+-- -------+