
Subject: Re: Clever Where() test
Posted by [thompson](#) on Thu, 25 May 1995 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

afl@cdc.noaa.gov (Andy Loughe) writes:

```
> In article <moleD93CEG.79n@netcom.com>, mole@netcom.com (Aaron Birenboim) writes:
> |> Is there a clever way to test for null return (-1) from WHERE???
> |>
> |> I'm doing stuff like :
> |>
> |>   i = WHERE(x GT 0.0)
> |>   s = SIZE(i)
> |>   IF s(0) EQ 0 THEN BEGIN
> |>     IF i GE 0 THEN BEGIN
> |>       ; do something with x(i)
> |>     ENDIF
> |>   ENDIF ELSE BEGIN
> |>     ; do something with the multiple elements x(i)
> |>   ENDIF
> |>
> |> Is there a more clever way to handle this? Especially since the
> |> "do something..." code is often the same whether i is scalar or vector.
> |>
> |> I wish that "IF i eq -1 THEN..." wouldnt crash for vector i's!!!
> |> --
```

> How about this?

```
> i = WHERE( x GT 0.0, count)
> IF ( count gt 0 ) then 'do something with the x's
```

> The count is very handy indeed. You could also build a
> loop from 0, count-1 if you needed to.

For that matter, one could also do

```
i = WHERE(x GT 0.0)
IF i(0) NE -1 THEN 'do something with the x's
```

although I also prefer using the optional count parameter. On the other hand,
if all one cared about was whether and matches were found, and didn't care
where they are, then one could do something like

```
IF (WHERE(x GT 0.0))(0) NE -1 THEN PRINT, 'At least one is > 0.0'
```

One can always reference a variable with (0), even if it's a scalar.

Bill Thompson
