
Subject: The limits of !NULL

Posted by [Jeremy Bailin](#) on Thu, 18 Apr 2013 02:52:42 GMT

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

Having only recently started switching to IDL8 thinking, I decided today to try out the WHERE(/NULL) functionality. And the first thing I tried to do failed miserably.

I think the fundamental problem is that !NULL doesn't seem to be at all useful on the RHS of an equation. What I wanted was to shorten:

```
a = where(array eq value, na)
if na gt 0 then begin
    data1[a] = -data1[a]
    data2[a] = -data2[a]
endif
```

to:

```
a = where(array eq value, /null)
data1[a] = -data1[a]
data2[a] = -data2[a]
```

since I thought the general idea of this functionality was so that you no longer had to test for no matches. However, when the condition is not met, it fails miserably:

```
IDL> data1[a] = -data1[a]
% Variable is undefined: <UNDEFINED>.
IDL> help, a
A          UNDEFINED = !NULL
```

If a is used only on the LHS, it works fine:

```
IDL> data1[a] = 5
IDL>
```

But even if the RHS is implicit, it fails:

```
IDL> data1[a]++
% Variable is undefined: A.
```

So I guess I'm sticking with the original coding paradigm.

-Jeremy.
