
Subject: Re: Efficient comparison of array location in two lists
Posted by [Heinz Stege](#) on Tue, 22 Feb 2011 02:45:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Mon, 21 Feb 2011 15:59:57 -0800 (PST), Matt Francis wrote:

> I have two sets of data, both comprised of a STRARR labelling the data
> and a FLTARR of the data itself.
>
> I want to compare the data in the two sets, but the order and number
> of labels are different, so I'd first need to find which array
> elements in each FLTARR correspond to the same label.
>
> It's trivial to see how this can be accomplished by simply looping
> over either data set and using WHERE(STRCMP()), but is there a way to
> do this without the loop?
>
> To make it clear, say I had:
>
> foo1 = ['a','b','c']
> foo2 = ['a','c','b']
>
> I want a single line that produces:
>
> IDL_MAGIC(foo1,foo2,indx1,indx2)
> print,indx1,indx2
> -> [0,1,2] [0,2,1]
>
> So that could compare
>
> foo1_data[indx1] and foo2_data[indx2] in whatever way I wanted.
>
> Any thoughts?

You can manage it in a short and very fast way by use of the built-in
function value_locate:

```
indx1=sort(foo1)
if n_elements(foo1) ge 2 then $
    indx2=value_locate(foo1[indx1],foo2) $
else $
    ; value locate does not work for vectors with 1 element.
    ; It would be nice, if ITT VIS could make it working in IDL 8.1.
    indx2=lonarr(n_elements(foo2))
ii=where(foo1[indx1] eq foo2[indx2],count)
if count le 0 then message,'No matching labels found.'
indx1=indx1[ii]
indx2=indx2[ii]
```

The labels foo1 as well as foo2 are assumed to be unique.

I believe, value_locate is an often underestimated function.

Heinz
