Subject: Re: Efficient comparison of array location in two lists
Posted by Heinz Stege on Tue, 22 Feb 2011 10:16:22 GMT
View Forum Message <> Reply to Message

On Mon, 21 Feb 2011 20:02:30 -0800 (PST), wlandsman wrote:

> On Monday, February 21, 2011 9:45:50 PM UTC-5, Heinz Stege wrote:
>
>> The labels foo1 as well as foo2 are assumed to be unique.
>
> I believe that this method also requires that foo2 be a one-to-one mapping of the elements in foo1, and fails for example, to find the matching 'e' values in the two vectors:
>
> foo1 = ['b','c','d','e']
> foo2 = ['a','e','f','g']
>
>> I believe, value_locate is an often underestimated function.
>
> And even some of us who use it often, forget that it can be used with strings. --Wayne

Good morning!

I should not answer to postings in the middle of the night. I copied
the commands from an existing program in my library and made mistakes
addopting them to the OPs question. Thank you Wayne, for bringing it
out.

And here is the correction:

```
indx1=sort(foo1)
if n_elements(foo1) ge 2 then $
  indx1=indx1[value_locate(foo1[indx1],foo2)] $
else $
  indx1=0
indx2=where(foo1[indx1] eq foo2,count)
if count le 0 then message,'No matching labels found.'
indx1=indx1[indx2]
```

Note, that you have to make changes, if you set the strictarrsubs or
idl2 compiler option. For this case:

```
indx1=sort(foo1)
if n_elements(foo1) ge 2 then $
  indx1=indx1[value_locate(foo1[indx1],foo2)>0] $
else $
  indx1=lonarr(n_elements(foo2))
indx2=where(foo1[indx1] eq foo2,count)
if count le 0 then message,'No matching labels found.'
```

indx1=indx1[indx2]

I hope, that it is working now. Sorry to everybody.

Heinz
