Subject: Re: Finding Common Elements in Two Arrays
Posted by landers on Fri, 03 Jun 1994 13:02:43 GMT
View Forum Message <> Reply to Message

I'm gonna rebut my own post.... a bit.

In article <1994Jun2.223011.12904@mksol.dseg.ti.com>, landers@tsunami.dseg.ti.com (David Landers) writes:

|> This is the first thing that poped into my head...
|>
|> Make 2-D arrays of A and B, like this:
|>
|> AA = A(*) # REPLICATE(1, N_ELEMENTS(B))
|> BB = REPLICATE(1, N_ELEMENTS(A)) # B(*)
|>

May be better to do:

```
Na = N_ELEMENTS(a)
Nb = N_ELEMENTS(b)
L = LINDGEN(Na,Nb)
AA = A( L MOD Na )
BB = B( L  /  Na )
```

|> Now you can compare all combinations of one to the other:
|>
|> ALIKE = UNIQUE( AA( WHERE( AA EQ BB ) ) )
|>
|> Then ALIKE contains a sorted list of the elements common to both arrays.

Yep. The advantage to the above over the matrix multiply (#) method is that it may be faster, and it will work with strings and structures (although EQ doesn't work with structs).

|> This could be more efficient, since you really only need a triangular array to
|> do this (you get 2 of each match, hence the UNIQUE).

What did I say? This is total nonsense (the triangular array bit). Just ignore this.

|>
|> --
|> Dave