
Subject: Re: Finding strings values common to two (large!) arrays

Posted by [Jeremy Bailin](#) on Wed, 28 Oct 2015 16:00:17 GMT

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

On Wednesday, October 7, 2015 at 11:21:31 AM UTC-4, [rj...@le.ac.uk](#) wrote:

> The IDs are of the form: 2009042300230430180019

>

> I think that's too long to convert into a number (at least when I try to turn it into a long it ends up very different!)

>

> CMSET_OP looks like it's what I need. Thanks both :-)

>

>

> On Wednesday, October 7, 2015 at 4:09:29 PM UTC+1, [wlandsman](#) wrote:

>> Two points to consider:

>>

>> I second Helder's suggestions but have two additional points to consider:

>>

>> 1. Do your array A have duplicate values? And if so, do you want to find the indices of all the values, even if they are repeated? Then I would suggest using

>>

>> <http://idlastro.gsfc.nasa.gov/ftp/pro/misc/match2.pro>

>>

>> which will return every matching index even of duplicate values.

>>

>> 2. You say the arrays are "numerical IDs in string format". Are you able to convert these strings into numerical values? If so, the matching algorithms work faster for numerical arrays (especially integers) than for strings. I do suspect the speed difference is not important unless you have to do the matching many times.

>>

>> --Wayne

>>

>> On Wednesday, October 7, 2015 at 10:45:55 AM UTC-4, [Helder](#) wrote:

>>> On Wednesday, October 7, 2015 at 4:13:59 PM UTC+2, [rj...@le.ac.uk](#) wrote:

>>>> I have arrays of numerical IDs in string format.

>>>>

>>>> I want to find all of the indices in Array A that contain a value that is present anywhere in Array B.

>>>>

>>>> The arrays are both quite large (>1 million values) so a loop is out of the question and them being strings complicates it as well.

>>>>

>>>> Any IDL Way tips?

>>>>

>>> Interesting... I guess that a set operation will do or in other words, you want to find (A) AND (B)

>>> Did you look at David's page:

>>> https://www.idlcoyote.com/tips/set_operations.html

```
>>>
>>> There are some good tips, among which Craig's CMSET_OP which works also on strings
(but does not return indices...).
>>>
>>> I hope it helps.
>>>
>>> Cheers,
>>> Helder
```

If you expect duplicates, this is a good example of when you might want to use `Value_Locate` as a mapping function (see http://www.idlcoyote.com/code_tips/valuelocate.html). First create a list of unique IDs:

```
unique_IDs = ID[uniq(ID, sort(ID))]
```

Then use them to map IDs from `array_A` (and B, etc):

```
mapped_ID = value_locate(unique_IDs, array_A)
```

`mapped_ID` has converted the strings into a densely-packed set of integers that can be used efficiently in set operations.

-Jeremy.
