Subject: Re: Saving for loop values into arrays without know the final size of array
Posted by Russell Ryan on Tue, 29 Jul 2014 19:58:00 GMT
View Forum Message <> Reply to Message

On Tuesday, July 29, 2014 3:24:15 PM UTC-4, sylveste...@gmail.com wrote:
> I am dealing with a large data set and a for-loop with some if statements involved. thus I do not know what the final size of the for-loop will be and cannot declare the array dimensions beforehand which allows you to save your for-loop answers into an array. Does anyone know of a suggestion about how to work around this?

Mike is right, if you have list, you should use list. However, this is will *NOT* work on IDL <7 and might be a problem for you (say if you distribute the code). If you want to use the "old school" way, then it's something like this.

You'll need to swap the for-loop for a while-loop and grow your arrays by concatenation. Suppose your array is called "arr" and you want it to be a floating point, then your code will look something like this:

```
arr=[0.0]
i=0L  ;init the counter
okay =1b  ;start the loop
while okay do begin
  if i eq 0 then begin
      arr(0)=FIRST_ELEMENT_OF_ARRAY
  endif else begin
      arr=[arr,ALL_OTHER_ELEMENTS_OF_ARRAY]
  endelse
  okay = SOME_TESTING_LOGIC_THAT_ONLY_YOU_KNOW
endwhile
```

This will work. You'll need to fill in the values of the array and stop condition, but we'd need to know more about your program to help. This is the simplest case and will work, however this can be made to be better in a great number of ways. First thing to do is change the line:

arr=[arr,XXXXX]

to something like

arr=[temporary(arr),XXXX]

this will save you memory and execute a tick faster (or always seems to).

Also, this is concatenation, which is horrifically slow in IDL, especially as the arrays get large. So, you're better off to concatenate as few times as possible. Instead of going through the relevant stuff, I'll point you to The Masters:

http://www.idlcoyote.com/code_tips/nullarray.html

Here, David Fanning details a discussion from JD Smith.

good luck...

Russell
