Subject: Re: Dynamically resizing arrays
Posted by netnews.comcast.net on Sun, 06 Feb 2005 04:08:28 GMT
View Forum Message <> Reply to Message

Adding data row by row like this can be a little inefficient as you are forcing IDL to allocate RAM for every iteration. I have found it to be a bit quicker to grow the array in chunks appropriate for your application.

Taking Andrew's example:

```
array_init=INTARR(100,100) ; allocate 100 rows initially
nrows = 0
totrows = 100

while (have_new_data = 1) do begin
    array_init[*,nrows] = your_row_of_data
    ++nrows

    ;  grow array if needed - adding 100 rows at a time
    if (nrows eq totrows - 1) then begin
       array_init = [[array_init],[INTARR(100,100)]]
       totrows = totrows + 100
     endif

endwhile

;  trim array when done adding data
array_init = array_init[*,nrows - 1]
```

You have a little more to keep track of but for larger arrays it will be worth the hassle.

-Rick

Andrew wrote:

> Hi Jonathan,
>
> Assume you have declared your initial array, lets call it array_init
>
> array_init=INTARR(100) ;for arguments sake
>
> if we now assume that you are in the loop and want to append the new
> data, which we also assume is 100 columns long (i.e fixed length)
>
> new_row=intarr(100) ;the array of new data

```
>  FOR i=0,99 DO BEGIN
>    ;some operation or whatever you do here
>    ;to place the data in new_row
>
>    array_init=[[array_init],[new_row]] ;appened it to the original data
>  ENDFOR
>
>  your array_init will now grow one row at a time with each loop. You
>  might want to consider using a WHILE statement though to avoid the FOR
>  loop. I hope this helps, and is correct. Try it with some dummy arrays
>  (I did).
>
>  Cheers
>  Andrew
>
>  Jonathan Greenberg wrote:
>
>> I was hoping to get some feedback on the best way of creating a
>
>  "database"
>
>> -- an array of fixed columns but unknown number of rows which will be
>> appended to within some sort of loop.  What is the best way of doing
>
>  this?
>
>> --j
>
>
```