Subject: Re: Where and lists of regions
Posted by Chris Lee on Mon, 11 Oct 2004 14:47:04 GMT
View Forum Message <> Reply to Message

In article <2svdmrF1pveu8U1@uni-berlin.de>, "Ben Panter" <me@privacy.net>
wrote:

> Hi,
>
> I'm in the middle of trying to simplify a large body of code which I
> inherited a few years ago and have been mangling ever since. Among other
> things, I'd be interested in anyone's thoughts on simplfying the case
> statement I use (full code below). Basically I have a list of regions
> which I want to remove from a vector. At the moment I use CASE to call
> WHERE to remove the regions elements (case to choose how many where
> statements are required). This is fine for a few regions, but I'm sure
> there must be a neater way, as when I need 10 regions, or even 20, this
> code is going to look terrible...
> I've written a code with EXECUTE, but I may want to use VM in the future
> so I'd like to avoid it if possible.
> Many Thanks,
>
> Ben

Hi,
I'm not sure this counts as 'neater' (and there are probably better
ways), but the following code does what you want, without EXECUTE. It
should be a drop in replacement for the CASE block of code. It works at
least for the trivial case of wave=findgen(1000)*10.

Chris.

```
for i=0,n_region-1 do em[i,*]=strsplit(em_list[i], '-', /extract)

;setup
em=reform(transpose(em), n_elements(em))
offset=1
endoffset=1

if(em[0] gt wave[0]) then begin
   em=[wave[0],em]
   offset=0
endif

if(em[n_elements(em)-1] le wave[n_elements(wave)-1]) then begin
```

```
    em=[em,wave[n_elements(wave)-1]]
endoffset=0
endif


w=value_locate(wave,em)
;bounds forcing
w=w>0 < n_elements(wave)-1

;special case of 'delete all'
if(offset and endoffset and n_elements(em) eq 4) then return ,-1

n=n_elements(w)-2+offset-endoffset ;take care of the special end cases

for i=offset, n,2 do begin
    if(n_elements(output) eq 0) then $ ;does output exist?no
      if(w[i] ge w[i+1]) then $ ;are the indices coincident? yes
      output=w[i+1] $
    else $                  ;no
      output=lindgen(w[i+1]-w[i]+1)+w[i]+1 $
    else $                  ;does output exit? yes
      if(w[i]+1 ge w[i+1]) then $ ;are the indices coincident? yes
      output=[output,w[i+1]] $
    else $                  ;no
      output=[output,lindgen(w[i+1]-w[i]+1)+w[i]+1]
endfor

if(offset eq 1) then output=[0,output];this, I think, is wrong, but it matches the original
```