
Subject: Re: IDLgrLegend Property Sheets, array properties

Posted by [sdettrick](#) on Wed, 21 Apr 2004 04:20:20 GMT

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

"Chris Torrence" <chris_torrence@NO-SPAM.yahoo.com> wrote in message
news:<108atgenh7rqdb3@corp.supernews.com>...

> Hi Sean,

Hi Chris,

thanks for the response. It is Very nice to hear from you folks at
RSI because one can feel comfortable that it is a definitive answer!

> You basically have two options for handling non-scalar properties:

>

> 1. You can make it a type USERDEF, create an ::EditUserdefProperty method,
> and then handle the property on your own, using some sort of custom widget.

> This is good for really complicated properties.

>

> 2. However, as you suggested, you can also just split the property up into
> multiple properties.

After tooling around for a while with _REF_EXTRA in GetProperty I
finally capitulated and went the way of your method 2. I like your
suggestion 1 though, it sounds much prettier. At present I have 20
hard-coded MY_ITEM_NAME_? and MY_ITEM_COLOR_? keywords (?=0-9) in
GetProperty. Nasty looking, but it works.

This is another case where the completely opaque nature of _REF_EXTRA
is a killer. Do you think there will ever be a function to fix that,
or do you know of a workaround?

In contrast the _EXTRA keyword is very friendly, and no hard-coded
list of keywords was needed in SetProperty. Instead it was
straightforward to strip out all of the tags, query ITEM_NAME with
GetProperty, get the index of the item_name passed as a keyword,
update the ITEM_NAME array with SetProperty, and there you go,
Robert's your father's brother(!):

```
; This SetProperty method accepts from  
; MY_ITEM_NAME_0 to MY_ITEM_NAME_32767,  
; via the _EXTRA keyword:
```

```
PRO myLegend::SetProperty, _EXTRA=EXTRA  
self->IDLgrLegend::SetProperty, _EXTRA=extra  
IF n_elements( extra ) ne 0 then begin  
    TAGS = tag_names( EXTRA )
```

```
name_locs = where( strmatch( TAGS, 'MY_ITEM_NAME_' ) eq 1 )
IF name_locs[0] ne -1 then begin
  self -> IDLgrLegend::GetProperty, ITEM_NAME=item_names
  FOR n=0,n_elements(name_locs)-1 do begin
    index = fix( strmid( tags[name_locs[n]], $
      strlen('MY_ITEM_NAME_'))
    item_names[index] = extra.(name_locs[n])
  ENDFOR
  self -> IDLgrLegend::SetProperty, ITEM_NAME=item_names
ENDIF
ENDIF
END
```

> Or, a hacky solution would be to limit the # of legend items to say 10, and
> then just hardcode a bunch of keywords to your subclass Get/Setproperty.

Yes indeed. Anyway now I have a Legend class that manages its own
property sheet widget so I can't complain too much.

> Hope this helps.

It helps a lot. Thanks a bundle.

Sean
