Subject: Re: MAKE_ARRAY question
Posted by btt on Fri, 13 Apr 2001 17:08:17 GMT
View Forum Message <> Reply to Message

Hi Paul,

X = MAKE_ARRAY(n_pts, TYPE = 4 + KEYWORD_SET(DOUBLE), /NOZERO)

The above should get you the correct type of array (Double type is 5
while Float type is 4.) I don't know if you can get into trouble with
simply passing the DOUBLE keyword along. It works on this MAC (a float
is returned if DOUBLE = 0).

Ben

Paul van Delst wrote:
>
> Hi there,
>
> I want to create an array that by default should be FLOAT but can be specified as DOUBLE
> by the user as a keyword input to a function. Instead of the following:
>
> FUNCTION my_func, n_pts, double = double
>
> IF (KEYWORD_SET(double)) THEN $
> x = MAKE_ARRAY(n_pts, /DOUBLE, /NOZERO) $
> ELSE $
> x = MAKE_ARRAY(n_pts, /FLOAT, /NOZERO)
>
> .....
>
> END
>
> can I rely on the fact that NO type keywords to MAKE_ARRAY means default FLOAT? I.e. :
>
> FUNCTION my_func, n_pts, double = double
>
> x = MAKE_ARRAY(n_pts, DOUBLE = double, /NOZERO)
>
> .....
>
> END
>
> The MAKE_ARRAY documentation does not specifically state that the default array created is
> a FLOAT. Does anyone know if this *is* documented anywhere so I'm not setting my function
> up (which I use *a lot*) to fall in a heap when IDL 5.5+ is released and RSI happened to
> decide a default LONG would be better?
>

> Or is there some other natty way to do this?
>
> Thanks for any help.
>
> paulv
>
> --
> Paul van Delst A little learning is a dangerous thing;
> CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
> Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
> Fax:(301)763-8545 And drinking largely sobers us again.
> paul.vandelst@noaa.gov Alexander Pope.

--
Ben Tupper
Bigelow Laboratory for Ocean Sciences
180 McKown Point Rd.
W. Boothbay Harbor, ME 04575
btupper@bigelow.org

Subject: Re: MAKE_ARRAY question
Posted by Paul van Delst on Fri, 13 Apr 2001 17:19:30 GMT
View Forum Message <> Reply to Message

Ben Tupper wrote:
>
> Hi Paul,
>
> X = MAKE_ARRAY(n_pts, TYPE = 4 + KEYWORD_SET(DOUBLE), /NOZERO)
>
> The above should get you the correct type of array (Double type is 5
> while Float type is 4.)

True, but if double = 3 (in which case the keyword is also considered "set") the type would be wrong (a string array!).

I don't know why users would do double = 3 in the call, but, in my experience at least, I've seen 'em do stranger things..... (me included of course - probably the worst offender actually :o).

> I don't know if you can get into trouble with
> simply passing the DOUBLE keyword along. It works on this MAC (a float
> is returned if DOUBLE = 0).

Works on my linux box too. I just want the behaviour blessed via some RSI documentation. Without that, I'm assuming it's pure luck.

Thanks,

paulv

--
Paul van Delst A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
Fax:(301)763-8545 And drinking largely sobers us again.
paul.vandelst@noaa.gov Alexander Pope.

Subject: Re: MAKE_ARRAY question
Posted by Paul van Delst on Fri, 13 Apr 2001 17:26:00 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:
>
> Hi there,
>
> I want to create an array that by default should be FLOAT but can be specified as DOUBLE
> by the user as a keyword input to a function. Instead of the following:
>
> FUNCTION my_func, n_pts, double = double
>
> IF (KEYWORD_SET(double)) THEN $
> x = MAKE_ARRAY(n_pts, /DOUBLE, /NOZERO) $
> ELSE $
> x = MAKE_ARRAY(n_pts, /FLOAT, /NOZERO)
>
> .....
>
> END
>
> can I rely on the fact that NO type keywords to MAKE_ARRAY means default FLOAT? I.e. :
>
> FUNCTION my_func, n_pts, double = double
>
> x = MAKE_ARRAY(n_pts, DOUBLE = double, /NOZERO)
>
> .....
>
> END
>
> The MAKE_ARRAY documentation does not specifically state that the default array created is
> a FLOAT. Does anyone know if this *is* documented anywhere so I'm not setting my function
> up (which I use *a lot*) to fall in a heap when IDL 5.5+ is released and RSI happened to
> decide a default LONG would be better?

>
> Or is there some other natty way to do this?

One could also do:

```
x = MAKE_ARRAY( n_pts, VALUE = 0.0, DOUBLE = double )
```

i.e value = float if double is not set, but I want to avoid the value assignation. The array can be quite large (many millions of points x 2) hence the /NOZERO. The performance hit of using the VALUE keyword is noticeable (the /NOZERO has no effect if VALUE is used).

paulv

--
Paul van Delst A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
Fax:(301)763-8545 And drinking largely sobers us again.
paul.vandelst@noaa.gov Alexander Pope.

Subject: Re: MAKE_ARRAY question
Posted by Craig Markwardt on Fri, 13 Apr 2001 18:38:10 GMT
View Forum Message <> Reply to Message

Paul van Delst <paul.vandelst@noaa.gov> writes:

> Ben Tupper wrote:
>>
>> Hi Paul,
>>
>> X = MAKE_ARRAY(n_pts, TYPE = 4 + KEYWORD_SET(DOUBLE), /NOZERO)
>>
>> The above should get you the correct type of array (Double type is 5
>> while Float type is 4.)
>
> True, but if double = 3 (in which case the keyword is also considered "set") the type
> would be wrong (a string array!).

Huh? So what if double EQ 3! Vis:

```
IDL> print, keyword_set(1)
      1
IDL> print, keyword_set(2)
      1
IDL> print, keyword_set(3)
      1
```

For KEYWORD_SET, I think "truth" is defined as non-zero.

Ooops. What do you think this one should report? Heh.

```
  print, keyword_set([0])
```

Craig

--

 -- --------------
Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 -- --------------

Subject: Re: MAKE_ARRAY question
Posted by Liam E. Gumley on Fri, 13 Apr 2001 18:48:03 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:
> One could also do:
>
>   x = MAKE_ARRAY(n_pts, VALUE = 0.0, DOUBLE = double)
>
> i.e value = float if double is not set, but I want to avoid the value assignation. The
> array can be quite large (many millions of points x 2) hence the /NOZERO. The performance
> hit of using the VALUE keyword is noticeable (the /NOZERO has no effect if VALUE is used).

```
if keyword_set(double) then $
  x = dblarr(npts, /nozero) else $
  x = fltarr(npts, /nozero)
```

OR

```
x = keyword_set(double) ? dblarr(npts, /nozero) : fltarr(npts, /nozero)
```

Cheers,
Liam.
http://cimss.ssec.wisc.edu/~gumley

Subject: Re: MAKE_ARRAY question
Posted by Liam E. Gumley on Fri, 13 Apr 2001 18:52:57 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:

> For KEYWORD_SET, I think "truth" is defined as non-zero.

From the IDL 5.3 documentation:
"The KEYWORD_SET function returns a nonzero value if Expression is
defined and nonzero or an array, otherwise zero is returned."

> Ooops. What do you think this one should report? Heh.
>
> print, keyword_set([0])

It should return 1, and it does:

IDL> print, keyword_set([0])
1

Cheers,
Liam.
http://cimss.ssec.wic.edu/~gumley/

Subject: Re: MAKE_ARRAY question
Posted by Paul van Delst on Fri, 13 Apr 2001 19:03:04 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:
>
> Paul van Delst <paul.vandelst@noaa.gov> writes:
>
>> Ben Tupper wrote:
>>>
>>> Hi Paul,
>>>
>>> X = MAKE_ARRAY(n_pts, TYPE = 4 + KEYWORD_SET(DOUBLE), /NOZERO)
>>>
>>> The above should get you the correct type of array (Double type is 5
>>> while Float type is 4.)
>>
>> True, but if double = 3 (in which case the keyword is also considered "set") the type
>> would be wrong (a string array!).
>
> Huh? So what if double EQ 3! Vis:
>
> IDL> print, keyword_set(1)
> 1
> IDL> print, keyword_set(2)
> 1
> IDL> print, keyword_set(3)
> 1

>
> For KEYWORD_SET, I think "truth" is defined as non-zero.

Yep - you're right. I was mistakenly assuming that the keyword value evaluated the same way as in the KEYWORD_SET true/false determination: even = false, odd = true, e.g.

```
IDL> if 1 then print, 'this is true'
this is true
IDL> if 2 then print, 'this is true'
IDL> if 3 then print, 'this is true'
this is true
IDL>
```

But I was kwrong.

Liam sent me a solution I like the best:

```
  x = keyword_set(double) ? dblarr(npts, /nozero) : fltarr(npts, /nozero)
```

yay and cool bananas.

paulv

--
Paul van Delst A little learning is a dangerous thing;
CIMSS @ NOAA/NCEP Drink deep, or taste not the Pierian spring;
Ph: (301)763-8000 x7274 There shallow draughts intoxicate the brain,
Fax:(301)763-8545 And drinking largely sobers us again.
paul.vandelst@noaa.gov Alexander Pope.