Subject: Re: How to add 'd' to get the correct julian conversion ?
Posted by David Fanning on Thu, 23 Jan 2003 16:19:45 GMT
View Forum Message <> Reply to Message

Kolbjorn Bekkelund (kolbjorn@arctic-linux.tnett.no) writes:

> How can I add the NEEDED d to get this:
>
> 2452662.305203d
>
> out of this:
> maxtime = jul2cal((data(0,maxgust_time)), /TO_STRING, /MDY)
>
> In my program (data(0,maxgust_time)) fetches 2452662.305203 out of the
> array, but if I don't add the d to the julian date it calculates the
> wrong time in the above statement.
>
> I've tried
>
> maxtime = jul2cal((data(0,maxgust_time))d, /TO_STRING, /MDY)
>
> but that's not accepted.

How about this:

maxtime = jul2cal(DOUBLE(data(0,maxgust_time)), /TO_STRING, /MDY)

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: How to add 'd' to get the correct julian conversion ?
Posted by Craig Markwardt on Thu, 23 Jan 2003 16:46:15 GMT
View Forum Message <> Reply to Message

Kolbjorn Bekkelund <kolbjorn@arctic-linux.tnett.no> writes:

> How can I add the NEEDED d to get this:
>
> 2452662.305203d
>

> out of this:
> maxtime = jul2cal((data(0,maxgust_time)), /TO_STRING, /MDY)
>
> In my program (data(0,maxgust_time)) fetches 2452662.305203 out of the
> array, but if I don't add the d to the julian date it calculates the
> wrong time in the above statement.

You can use
double(data(0,maxgust_time)),
but the variable DATA should already be in double precision. At least
it should be if you expect 13 decimal digits of precision to be
maintained. When you type the number directly on the command line,
you probably do have to use the "D" to indicate double precision, but
you should not have to if the variable DATA is already double.

Craig

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

Subject: Re: How to add 'd' to get the correct julian conversion ?
Posted by Kolbjorn Bekkelund on Thu, 23 Jan 2003 20:02:26 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:
> Kolbjorn Bekkelund <kolbjorn@arctic-linux.tnett.no> writes:
>
>
>> How can I add the NEEDED d to get this:
>>
>> 2452662.305203d
>>
>> out of this:
>> maxtime = jul2cal((data(0,maxgust_time)), /TO_STRING, /MDY)
>>
>> In my program (data(0,maxgust_time)) fetches 2452662.305203 out of the
>> array, but if I don't add the d to the julian date it calculates the
>> wrong time in the above statement.
>
>
> You can use
> double(data(0,maxgust_time)),
> but the variable DATA should already be in double precision. At least
> it should be if you expect 13 decimal digits of precision to be

> maintained. When you type the number directly on the command line,
> you probably do have to use the "D" to indicate double precision, but
> you should not have to if the variable DATA is already double.
>
> Craig
>

I've checked my array a bit more and it seems as if there's something
wrong with it. From the file I'm reading in with read-ascii I should
have this:
2452662.499876 2.719500 6.216000 343.494000
955.793400 93.911600 -5.444307

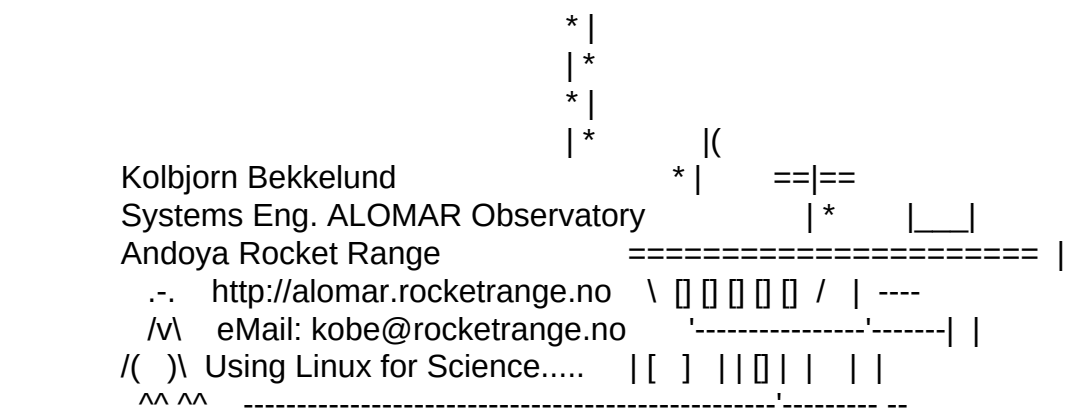
but the print, data in IDL shows:
2.45266e+06 2.71950 6.21600 343.494 955.793
93.9116 -5.44431

If I replace the read-acsii with Reimar Bauers read_data_file I get:
2452662.5 2.7195000 6.2160000 343.49400
955.79340 93.911600 -5.4443070

but as you see the julian date in the first element is wrong in both
arrays. How can I do ensure that I get all digits inserted ?

Kolbjorn

--



Subject: Re: How to add 'd' to get the correct julian conversion ?
Posted by David Fanning on Thu, 23 Jan 2003 22:15:46 GMT

View Forum Message <> Reply to Message

Kolbjorn Bekkelund (kolbjorn@arctic-linux.tnett.no) writes:

> I've checked my array a bit more and it seems as if there's something

> wrong with it. From the file I'm reading in with read-ascii I should
> have this:
> 2452662.499876 2.719500 6.216000 343.494000
> 955.793400 93.911600 -5.444307
>
> but the print, data in IDL shows:
> 2.45266e+06 2.71950 6.21600 343.494 955.793
> 93.9116 -5.44431
>
> If I replace the read-acsii with Reimar Bauers read_data_file I get:
> 2452662.5 2.7195000 6.2160000 343.49400
> 955.79340 93.911600 -5.4443070
>
> but as you see the julian date in the first element is wrong in both
> arrays. How can I do ensure that I get all digits inserted ?

Ah, yes, I suspected this might be the problem earlier, but
I didn't have time to respond properly. I think you will
be interested in this article:

http://www.dfanning.com/math_tips/sky_is_falling.html

The specific answer to your question is to read your
data into a double-precision variable. Since I've never
used READ_ASCII I can't tell you how to do this, although
I presume there must be a way. :-(

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: How to add 'd' to get the correct julian conversion ?
Posted by Paul Van Delst[1] on Thu, 23 Jan 2003 22:20:07 GMT
View Forum Message <> Reply to Message

Kolbjorn Bekkelund wrote:
>
> Craig Markwardt wrote:
>> Kolbjorn Bekkelund <kolbjorn@arctic-linux.tnett.no> writes:
>>
>>

>>> How can I add the NEEDED d to get this:
>>>
>>> 2452662.305203d
>>>
>>> out of this:
>>> maxtime = jul2cal((data(0,maxgust_time)), /TO_STRING, /MDY)
>>>
>>> In my program (data(0,maxgust_time)) fetches 2452662.305203 out of the
>>> array, but if I don't add the d to the julian date it calculates the
>>> wrong time in the above statement.
>>
>>
>> You can use
>>   double(data(0,maxgust_time)),
>> but the variable DATA should already be in double precision. At least
>> it should be if you expect 13 decimal digits of precision to be
>> maintained. When you type the number directly on the command line,
>> you probably do have to use the "D" to indicate double precision, but
>> you should not have to if the variable DATA is already double.
>>
>> Craig
>>
>
> I've checked my array a bit more and it seems as if there's something
> wrong with it. From the file I'm reading in with read-ascii I should
> have this:
> 2452662.499876 2.719500   6.216000   343.494000
> 955.793400   93.911600   -5.444307
>
> but the print, data in IDL shows:
> 2.45266e+06   2.71950   6.21600   343.494   955.793
> 93.9116   -5.44431
>
> If I replace the read-acsii with Reimar Bauers read_data_file I get:
> 2452662.5   2.7195000   6.2160000   343.49400
> 955.79340   93.911600   -5.4443070
>
> but as you see the julian date in the first element is wrong in both
> arrays. How can I do ensure that I get all digits inserted ?

If I understand your question (and I'm not sure I do) there _may_ be two things going on here. David Fanning's post addresses one thing. The other is, if you want to *see* all the digits on screen, then you must use a format string. A generic "print, data" statements means that IDL prints out what the IDL-writers thought was a reasonable number of digits for the data type. If you want something other than the default (i.e. 6 sig figs for your apparently single precision floating points above), you gotta specify it explicitly using

  print, format='(<some useful format descriptor>)', data

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC
Ph: (301)763-8000 x7274
Fax:(301)763-8545

Subject: Re: How to add 'd' to get the correct julian conversion ?
Posted by thompson on Thu, 23 Jan 2003 23:38:50 GMT
View Forum Message <> Reply to Message

Kolbjorn Bekkelund <kolbjorn@arctic-linux.tnett.no> writes:

> Craig Markwardt wrote:
>> Kolbjorn Bekkelund <kolbjorn@arctic-linux.tnett.no> writes:
>>
>>
>>> How can I add the NEEDED d to get this:
>>>
>>> 2452662.305203d
>>>
>>> out of this:
>>> maxtime = jul2cal((data(0,maxgust_time)), /TO_STRING, /MDY)
>>>
>>> In my program (data(0,maxgust_time)) fetches 2452662.305203 out of the
>>> array, but if I don't add the d to the julian date it calculates the
>>> wrong time in the above statement.
>>
>>
>> You can use
>> double(data(0,maxgust_time)),
>> but the variable DATA should already be in double precision. At least
>> it should be if you expect 13 decimal digits of precision to be
>> maintained. When you type the number directly on the command line,
>> you probably do have to use the "D" to indicate double precision, but
>> you should not have to if the variable DATA is already double.
>>
>> Craig
>>

> I've checked my array a bit more and it seems as if there's something
> wrong with it. From the file I'm reading in with read-ascii I should
> have this:
> 2452662.499876 2.719500 6.216000 343.494000

> 955.793400 93.911600 -5.444307

> but the print, data in IDL shows:
> 2.45266e+06 2.71950 6.21600 343.494 955.793
> 93.9116 -5.44431

> If I replace the read-acsii with Reimar Bauers read_data_file I get:
> 2452662.5 2.7195000 6.2160000 343.49400
> 955.79340 93.911600 -5.4443070

> but as you see the julian date in the first element is wrong in both
> arrays. How can I do ensure that I get all digits inserted ?

I tried the following

```
IDL> a = 2452662.499876  ;Single precision
IDL> print,a
  2.45266e+06
IDL> a = 2452662.499876d ;Double precision
IDL> print,a
       2452662.5
```

It looks like read_data_file is reading the data correctly as double precision, while read_ascii is apparently reading everything into as single precision. I know that it looks like A is being rounded off in the second case, but that's only because of the default format being used for printing. If you use an explicit format, you can see more of the digits.

```
IDL> print,a,format='(F20.6)'
      2452662.499876
```

If, on the other hand, the data was read in as single precision, it really will be truncated.

```
IDL> a = 2452662.499876  ;Single precision
IDL> print,a,format='(F20.6)'
      2452662.500000
```

Bill Thompson

Subject: Re: How to add 'd' to get the correct julian conversion ?
Posted by R.Bauer on Fri, 31 Jan 2003 18:54:22 GMT
View Forum Message <> Reply to Message

Kolbjorn Bekkelund wrote:

> Craig Markwardt wrote:

>> Kolbjorn Bekkelund <kolbjorn@arctic-linux.tnett.no> writes:
>>
>>
>>> How can I add the NEEDED d to get this:
>>>
>>> 2452662.305203d
>>>
>>> out of this:
>>> maxtime = jul2cal((data(0,maxgust_time)), /TO_STRING, /MDY)
>>>
>>> In my program (data(0,maxgust_time)) fetches 2452662.305203 out of the
>>> array, but if I don't add the d to the julian date it calculates the
>>> wrong time in the above statement.
>>
>>
>> You can use
>>   double(data(0,maxgust_time)),
>> but the variable DATA should already be in double precision. At least
>> it should be if you expect 13 decimal digits of precision to be
>> maintained. When you type the number directly on the command line,
>> you probably do have to use the "D" to indicate double precision, but
>> you should not have to if the variable DATA is already double.
>>
>> Craig
>>
>
> I've checked my array a bit more and it seems as if there's something
> wrong with it. From the file I'm reading in with read-ascii I should
> have this:
> 2452662.499876 2.719500   6.216000   343.494000
> 955.793400   93.911600   -5.444307
>
> but the print, data in IDL shows:
> 2.45266e+06   2.71950   6.21600   343.494   955.793
> 93.9116   -5.44431
>
> If I replace the read-acsii with Reimar Bauers read_data_file I get:
> 2452662.5   2.7195000   6.2160000   343.49400
> 955.79340   93.911600   -5.4443070
>
> but as you see the julian date in the first element is wrong in both
> arrays. How can I do ensure that I get all digits inserted ?

Dear Kolbjorn

The problem I think you have is that's the default format for print is
defined for float numbers.
read_data_file uses as default double if you don't give a type.

So you should try something like

```
 x=read_data_file('test.dat')
 print, x.data[0],format='(F20.10)'
  2452662.4998760000
```

best regards

Reimar

>
> Kolbjorn
>
>

--
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
http://www.fz-juelich.de/icg/icg-i/
 ========== ======
a IDL library at ForschungsZentrum Juelich
 http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro. html