Subject: Double precision data into caldat
Posted by laura.hike on Tue, 08 Sep 2015 19:37:23 GMT
View Forum Message <> Reply to Message

Hi,

I'm trying to convert some Julian dates back into standard dates using caldat. (In fact, I am testing what I got out of julday in the first place.) There should be hours and minutes in the results. If I take the output of julday as a variable, say TEST, and put it into caldat as

```
caldat, test, m, d, y, h, mm, s
```

I get the right answer. However, if I just use the actual value of test, I get odd results. So, for the Julian day 2456658.56250000, I should get

```
2014        1        1        1       30       0
```

in year, month, day, hour, min, sec form. If I use

```
caldat, 2456658.56250000D, m, d, y, h, mm, s
```

the results are correct, but if I do the type conversion using double(), it doesn't. So I have

```
caldat, double(2456658.56250000), m, d, y, h, mm, s
```

and

```
p = double(2456658.56250000)
caldat, p, m, d, y, h, mm, s
```

both giving the result

```
2014        1        1        0        0      0
```

Does anyone know why this is? As far as I know, all of those input values are the same.

Thanks!

PS Why on earth does IDL use month, day, year rather than year, month, day in both caldat and julday??

Subject: Re: Double precision data into caldat
Posted by David Fanning on Tue, 08 Sep 2015 20:01:05 GMT
View Forum Message <> Reply to Message

Larry H. writes:

> I'm trying to convert some Julian dates back into standard dates using caldat. (In fact, I am testing what I got out of julday in the first place.) There should be hours and minutes in the results. If I take the output of julday as a variable, say TEST, and put it into caldat as
>
> caldat, test, m, d, y, h, mm, s
>
> I get the right answer. However, if I just use the actual value of test, I get odd results. So, for the Julian day 2456658.56250000, I should get
>
> 2014 1 1 1 30 0
>
> in year, month, day, hour, min, sec form. If I use
>
> caldat, 2456658.56250000D, m, d, y, h, mm, s
>
> the results are correct, but if I do the type conversion using double(), it doesn't. So I have
>
> caldat, double(2456658.56250000), m, d, y, h, mm, s
>
> and
>
> p = double(2456658.56250000)
> caldat, p, m, d, y, h, mm, s
>
> both giving the result
>
> 2014 1 1 0 0 0
>
> Does anyone know why this is? As far as I know, all of those input values are the same.

Some required reading for all those who work with IDL:

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

> PS Why on earth does IDL use month, day, year rather than year, month, day in both caldat and julday??

I believe this was an attempt to make IDL "unique" among programming languages. It has infuriated some and made IDL endearing to others. ;-)

Cheers,

David
--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: Double precision data into caldat
Posted by laura.hike on Tue, 08 Sep 2015 20:38:35 GMT
View Forum Message <> Reply to Message

On Tuesday, September 8, 2015 at 1:01:09 PM UTC-7, David Fanning wrote:

>
>> PS Why on earth does IDL use month, day, year rather than year, month, day in both caldat and julday??
>
> I believe this was an attempt to make IDL "unique" among programming
> languages. It has infuriated some and made IDL endearing to others. ;-)
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

All I know is that it causes me to make a lot of mistakes!

Subject: Re: Double precision data into caldat
Posted by laura.hike on Tue, 08 Sep 2015 20:43:28 GMT
View Forum Message <> Reply to Message

On Tuesday, September 8, 2015 at 1:01:09 PM UTC-7, David Fanning wrote:

> Larry H. writes:
>
>> I'm trying to convert some Julian dates back into standard dates using
> caldat. (In fact, I am testing what I got out of julday in the first
> place.) There should be hours and minutes in the results. If I take
> the output of julday as a variable, say TEST, and put it into caldat as
>>
>> caldat, test, m, d, y, h, mm, s
>>

>> I get the right answer. However, if I just use the actual value of test, I get odd results. So, for the Julian day 2456658.56250000, I should get
>>
>> 2014 1 1 1 30 0
>>
>> in year, month, day, hour, min, sec form. If I use
>>
>> caldat, 2456658.56250000D, m, d, y, h, mm, s
>>
>> the results are correct, but if I do the type conversion using double(), it doesn't. So I have
>>
>> caldat, double(2456658.56250000), m, d, y, h, mm, s
>>
>> and
>>
>> p = double(2456658.56250000)
>> caldat, p, m, d, y, h, mm, s
>>
>> both giving the result
>>
>> 2014 1 1 0 0 0
>>
>> Does anyone know why this is? As far as I know, all of those input values are the same.
>
> Some required reading for all those who work with IDL:
>
> http://www.idlcoyote.com/math_tips/sky_is_falling.html
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

PS I looked at the web page you mentioned, but it doesn't seem to help. It doesn't take 15 significant digits to express hours as fractional days, which can be seen from the original value I'm working with. What I'm trying to find out is why putting a D after a number is different than applying the double function. As far as I can tell, caldat isn't recognizing the value as a double and just returning the date using integers.

Subject: Re: Double precision data into caldat
Posted by Lajos Foldy on Tue, 08 Sep 2015 20:54:54 GMT

View Forum Message <> Reply to Message

On Tuesday, September 8, 2015 at 9:37:25 PM UTC+2, Larry H. wrote:

> As far as I know, all of those input values are the same.

False.

```
IDL> print, format='(F20.10)', 2456658.56250000D
  2456658.5625000000
IDL> print, format='(F20.10)', 2456658.56250000
  2456658.5000000000
IDL> print, format='(F20.10)', double(2456658.56250000)
  2456658.5000000000
```

(2456658.5625000000 is a float, with ~7 digits precision.)

regards,
Lajos

> Thanks!
>
>
>
> PS Why on earth does IDL use month, day, year rather than year, month, day in both caldat and julday??

Subject: Re: Double precision data into caldat
Posted by Paul Van Delst[1] on Tue, 08 Sep 2015 20:58:46 GMT
View Forum Message <> Reply to Message

Hello,

On 09/08/15 15:37, Larry H. wrote:
> Hi,
>
> I'm trying to convert some Julian dates back into standard dates
> using caldat. (In fact, I am testing what I got out of julday in the
> first place.) There should be hours and minutes in the results. If
> I take the output of julday as a variable, say TEST, and put it into
> caldat as
>
> caldat, test, m, d, y, h, mm, s
>
> I get the right answer. However, if I just use the actual value of
> test, I get odd results. So, for the Julian day 2456658.56250000, I

> should get
>
> 2014 1 1 1 30 0
>
> in year, month, day, hour, min, sec form. If I use
>
> caldat, 2456658.56250000D, m, d, y, h, mm, s
>
> the results are correct, but if I do the type conversion using
> double(), it doesn't. So I have
>
> caldat, double(2456658.56250000), m, d, y, h, mm, s
>
> and
>
> p = double(2456658.56250000) caldat, p, m, d, y, h, mm, s
>
> both giving the result
>
> 2014 1 1 0 0 0
>
> Does anyone know why this is? As far as I know, all of those input
> values are the same.

Nope. They are not.

```
IDL> p=2456658.56250000
IDL> help, p
P               FLOAT     =   2.45666e+06
IDL> print, p, format='(f25.10)'
       2456658.5000000000
```

Single precision has about 7-8 significant digits.

So for a value you write as "2456658.56250000" -- which is a single
precision literal constant -- you've only got "2456658.5" of useful digits.

When you use DOUBLE, as in:

```
IDL> p=DOUBLE(2456658.56250000)
```

you are taking a SINGLE PRECISION literal constant (2456658.56250000)
and converting it to double. So you still lose the extra information
(the "X.X625") stuff, e.g.

```
IDL> p=DOUBLE(2456658.56250000)
IDL> help, p
P               DOUBLE    =       2456658.5
```

```
IDL> print, p, format='(f25.10)'
       2456658.5000000000
```

Now, if you declare a double precision literal constant, like so:

```
IDL> p=2456658.56250000d0
IDL> help, p
P               DOUBLE    =       2456658.6
```

you are "declaring" a double precision literal so you have about 15
digits of precision-y goodness which you see when you print it out:

```
IDL> print, p, format='(f25.10)'
       2456658.5625000000
```

Basically, if you ever declare literal constants in code where those
values can have more than 8 useful digits, always use double precision
declarations. Converting a single precision 8+ sig fig number to double
will have no effect.

Hope that helps.

cheers,

paulv

Subject: Re: Double precision data into caldat
Posted by Jim Pendleton on Tue, 08 Sep 2015 21:18:56 GMT
View Forum Message <> Reply to Message

On Tuesday, September 8, 2015 at 2:01:09 PM UTC-6, David Fanning wrote:
> Larry H. writes:
>
>> I'm trying to convert some Julian dates back into standard dates using
> caldat. (In fact, I am testing what I got out of julday in the first
> place.) There should be hours and minutes in the results. If I take
> the output of julday as a variable, say TEST, and put it into caldat as
>>
>> caldat, test, m, d, y, h, mm, s
>>
>> I get the right answer. However, if I just use the actual value of test, I get odd results. So, for the Julian day 2456658.56250000, I should get
>>
>> 2014 1 1 1 30 0
>>
>> in year, month, day, hour, min, sec form. If I use

>>
>>   caldat, 2456658.56250000D, m, d, y, h, mm, s
>>
>> the results are correct, but if I do the type conversion using double(), it doesn't. So I have
>>
>>   caldat, double(2456658.56250000), m, d, y, h, mm, s
>>
>> and
>>
>>   p = double(2456658.56250000)
>>   caldat, p, m, d, y, h, mm, s
>>
>> both giving the result
>>
>>   2014   1   1   0   0   0
>>
>> Does anyone know why this is? As far as I know, all of those input values are the same.
>
> Some required reading for all those who work with IDL:
>
>   http://www.idlcoyote.com/math_tips/sky_is_falling.html
>
>
>> PS Why on earth does IDL use month, day, year rather than year, month, day in both caldat and julday??
>
> I believe this was an attempt to make IDL "unique" among programming
> languages. It has infuriated some and made IDL endearing to others. ;-)
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

I'm going to recommend a recently published book by my colleague Ron Kneusel, "Numbers and Computers", published by Springer and available on Amazon. It's like the "Sky is Falling" essay, but on steroids.

Ron goes into great depth on the topic of how computers represent data such as integers, both big and small, fixed point and floating point numbers, rational numbers, etc.

He also discusses how mathematical operations from basic comparisons to trigonometric functions are frequently implemented. All this is described at a fundamental bit-twiddling level.

Though there are no IDL examples (what's up with that?) there are full algorithmic implementations in C, on which IDL is largely based, and frequently in Python as well.

Also see numbersandcomputers.com.

Jim P.

Subject: Re: Double precision data into caldat
Posted by laura.hike on Tue, 08 Sep 2015 22:22:52 GMT
View Forum Message <> Reply to Message

Yes, this answers the question. Thanks!

On Tuesday, September 8, 2015 at 1:58:48 PM UTC-7, Paul van Delst wrote:
> Hello,
>
> On 09/08/15 15:37, Larry H. wrote:
>> Hi,
>>
>> I'm trying to convert some Julian dates back into standard dates
>> using caldat. (In fact, I am testing what I got out of julday in the
>> first place.) There should be hours and minutes in the results. If
>> I take the output of julday as a variable, say TEST, and put it into
>> caldat as
>>
>> caldat, test, m, d, y, h, mm, s
>>
>> I get the right answer. However, if I just use the actual value of
>> test, I get odd results. So, for the Julian day 2456658.56250000, I
>> should get
>>
>> 2014   1   1   1   30   0
>>
>> in year, month, day, hour, min, sec form.   If I use
>>
>> caldat, 2456658.56250000D, m, d, y, h, mm, s
>>
>> the results are correct, but if I do the type conversion using
>> double(), it doesn't. So I have
>>
>> caldat, double(2456658.56250000), m, d, y, h, mm, s
>>
>> and
>>
>> p = double(2456658.56250000) caldat, p, m, d, y, h, mm, s
>>

>> both giving the result
>>
>> 2014 1 1 0 0 0
>>
>> Does anyone know why this is? As far as I know, all of those input
>> values are the same.
>
> Nope. They are not.
>
> IDL> p=2456658.56250000
> IDL> help, p
> P FLOAT = 2.45666e+06
> IDL> print, p, format='(f25.10)'
> 2456658.5000000000
>
> Single precision has about 7-8 significant digits.
>
> So for a value you write as "2456658.56250000" -- which is a single
> precision literal constant -- you've only got "2456658.5" of useful digits.
>
> When you use DOUBLE, as in:
>
> IDL> p=DOUBLE(2456658.56250000)
>
> you are taking a SINGLE PRECISION literal constant (2456658.56250000)
> and converting it to double. So you still lose the extra information
> (the "X.X625") stuff, e.g.
>
> IDL> p=DOUBLE(2456658.56250000)
> IDL> help, p
> P DOUBLE = 2456658.5
> IDL> print, p, format='(f25.10)'
> 2456658.5000000000
>
> Now, if you declare a double precision literal constant, like so:
>
> IDL> p=2456658.56250000d0
> IDL> help, p
> P DOUBLE = 2456658.6
>
> you are "declaring" a double precision literal so you have about 15
> digits of precision-y goodness which you see when you print it out:
>
> IDL> print, p, format='(f25.10)'
> 2456658.5625000000
>
>
> Basically, if you ever declare literal constants in code where those

> values can have more than 8 useful digits, always use double precision
> declarations. Converting a single precision 8+ sig fig number to double
> will have no effect.
>
> Hope that helps.
>
> cheers,
>
> paulv

Subject: Re: Double precision data into caldat
Posted by David Fanning on Tue, 08 Sep 2015 22:43:43 GMT
View Forum Message <> Reply to Message

Larry H. writes:

> Yes, this answers the question. Thanks!

As did the article, but, c'est la vie! ;-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")
