
Subject: Re: Conversion of MS 64bit timestamps to JD
Posted by [Dick Jackson](#) on Mon, 14 Nov 2016 16:09:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Andrew,

That looks promising, but do those dates/times seem at least roughly correct? With the calendar systems and time zone issues, I'd want to know that we're not a couple of days or hours off. (I'd implicitly trust the minutes and seconds, even the tenth-of-a-microseconds! :-)

I notice that your output is only giving microsecond precision. If you were at all concerned about that, I'd want to see a couple of zeroes to reassure that the conversion is good. I'm guessing that you're using Float for these values, but that is not enough to allow the 'seconds' to go up to 60 and allow seven accurate places after the decimal. So, if this matters to you, I'd use Double when computing seconds (at least) and print with eight or nine places, expecting to see some zeroes.

Thanks for the stickers. :-)

Cheers,
-Dick

On Sunday, 13 November 2016 03:48:19 UTC-8, andrew...@gmail.com wrote:

> Hi Dick,

>

> Using your method, here are the decodes of the 64bit timestamps
> embedded in a short SER image that a friend supplied.

>

> Cheers,

>

> Andrew

>

>

>	2016.0000	11.000000	13.000000	6.0000000	38.000000	16.072238
>	2016.0000	11.000000	13.000000	6.0000000	38.000000	16.332335
>	2016.0000	11.000000	13.000000	6.0000000	38.000000	16.593015
>	2016.0000	11.000000	13.000000	6.0000000	38.000000	16.851370
>	2016.0000	11.000000	13.000000	6.0000000	38.000000	17.114491
>	2016.0000	11.000000	13.000000	6.0000000	38.000000	17.456318
>	2016.0000	11.000000	13.000000	6.0000000	38.000000	17.633305
>	2016.0000	11.000000	13.000000	6.0000000	38.000000	17.891857
>	2016.0000	11.000000	13.000000	6.0000000	38.000000	18.152162
>	2016.0000	11.000000	13.000000	6.0000000	38.000000	18.412077
>	2016.0000	11.000000	13.000000	6.0000000	38.000000	18.671080
>	2016.0000	11.000000	13.000000	6.0000000	38.000000	18.932369
>	2016.0000	11.000000	13.000000	6.0000000	38.000000	19.191133
>	2016.0000	11.000000	13.000000	6.0000000	38.000000	19.451026

>

>

> On Sunday, 13 November 2016 09:55:34 UTC+10:30, andrew...@gmail.com wrote:
>> Hi Dick,
>>
>> This is Gold Star stuff! Maybe even an Elephant stamp too.
>>
>> Of course, having the ability to record 100nanosecond intervals assumes that the Windows clock is up to that int eh first place, and well calibrated, but that's another story.
>>
>> A friend and I recorded and occultation by the Centaur asteroid Chariklo
>> on Oct 1st, using SER format. I've written a converter to get individual FITS
>> files from the SER file, but it needs those 64bit timestamps converted too.
>>
>> Many Thanks.
>>
>> Andrew
>>
>> Working the other way, to turn serDateTime into six parameters:
>>
>> IDL> serDateTime = 636144921840000001 ; 2016-11-11T20:16:24.0000001 from above
>>
>> IDL> Jul2Greg,(serDateTime / (24LL * 60 * 60 * 10000000)) + Greg2Jul(1,1,1,0,0,0),mo,d,y
>> IDL> increments = serDateTime MOD (10000000LL * 24 * 60 * 60)
>> IDL> s = increments MOD (10000000LL * 60) / 1D7
>> IDL> m = increments / (10000000LL * 60) MOD 60
>> IDL> h = increments / (10000000LL * 60 * 60)
>>
>> This seems to have worked:
>>
>> IDL> help,y,mo,d,h,m
>> Y LONG = 2016
>> MO LONG = 11
>> D LONG = 11
>> H LONG64 = 20
>> M LONG64 = 16
>>
>>
>>> Andrew
