

---

Subject: Time conversion

Posted by [Alex Schuster](#) on Wed, 14 Mar 2001 16:52:50 GMT

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

---

Hi!

I have to convert some date formats, and I wonder why IDL does not give more support here. A common date/time format is the seconds since 1970, and the systime() funtion can return the current date in this format. Yippie! But why is there no function to convert this from/to standard ASCII date strings, or from/to Julian date?

I found such a thing in the Astro library, st2date, input is the seconds-since-1970, output is year, month, day etc., and even day and month in ASCII notation if one likes. Great. But I did not find something yet to convert to seconds-since-1970. I could write it myself, but I guess this already had be done some dozen times. Some big routine with many, many keywords to convert between all those formats would be nice :)

Alex

--

Alex Schuster    Wonko@weird.cologne.de  
alex@pet.mpin-koeln.mpg.de

PGP Key available

---

---

Subject: Re: Time conversion

Posted by [John-David T. Smith](#) on Wed, 14 Mar 2001 21:01:04 GMT

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

---

tam wrote:

>

> JD Smith wrote:

> ... In any case,

>> it is oh so trivial to convert date to systime, the inverse:

>>

>> st=(julday(month,day,year,hr,min,sec)-2440587.5D)\*86400.0D

>>

>> That's it! For instance:

>>

>> IDL> st0=systime(0) & st1=systime(1)

>> IDL> d=bin\_date(st0)

>> IDL> st=(julday(d[1],d[2],d[0],d[3],d[4],d[5])-2440587.5D)\*86400. 0D

>> IDL> print,FORMAT='(A,":",2D30.5)',st0,st1,st

>>

> ...

>>

>> JD

>  
> How are leap seconds accounted for here? They usually cause  
> niggling problems if one actually needs to worry about  
> precision at the second level over periods of more than a  
> month or two. It looks like your code is ignoring them.  
> The biggest pain is that they are not predictable so the  
> code needs to access an external file, or update an  
> internal table periodically.

As you can see, all of the details of time handling are left to internal IDL routines. Inasmuch as IDL may or may not deal with leap seconds, so too would this algorithm. That said, I'm quite certain they don't.

This is of course the reason to use Julian Date, which explicitly don't include leap seconds, and thus is quite continuous, independent of the whimsy of datekeepers. Currently there are about 1 minute 5 seconds between UTC (the physical time clocks keep, including leap seconds), and the Julian date. There have been rumours of cancelling the leap second program for a variety of reasons.

Good luck,

JD

---

Subject: Re: Time conversion  
Posted by [John-David T. Smith](#) on Tue, 20 Mar 2001 15:19:25 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Alex Schuster wrote:

>  
> JD Smith wrote:  
>  
>> Alex Schuster wrote:  
>  
>>> I found such a thing in the Astro library, st2date, input is the  
>>> seconds-since-1970, output is year, month, day etc., and even day and  
>>> month in ASCII notation if one likes. Great. But I did not find  
>>> something yet to convert to seconds-since-1970. I could write it myself,  
>>> but I guess this already had be done some dozen times. Some big routine  
>>> with many, many keywords to convert between all those formats would be  
>>> nice :)  
>>  
>> Hmm... I doubt you found that in the Astro library, since I wrote it and  
>> posted it to the newsgroup back in 1998. I did use daycnv from the  
>> astro package to convert julian to calendar dates.  
>  
> Umm, right. I found it in the folder where my copy of the Astro library

```

> resides.
>
>> The recent
>> IDL-bundled "caldat" will work just as well for that now.
>
> Hooray! I hope it also can output various time strings... but I'm still
> using IDL 5.2 :(
>
>> In any case,
>> it is oh so trivial to convert date to systime, the inverse:
>>
>> st=(julday(month,day,year,hr,min,sec)-2440587.5D)*86400.0D
>
> Okay, right. Once you know that these Julian Days are, and if you don't
> think about leap seconds. I already did the same, but made the mistake
> of using julday(1,1,1970) instead of julday(1,1,1970,0,0,0), which gives
> half a day offset. Okay, my fault, but if IDL already had such a
> conversion routine, I wouldn't have to think about it, and I wouldn't be
> able to make such mistakes.
> Another example, why can't systime() just also output julian days, give
> it one more keyword and that's it.

```

You mean like:

```

=====
=====

```

Keywords

JULIAN

Set this keyword to specify that the current time is to be returned as a  
a double precision floating value containing the current Julian  
date/time....

```

=====
=====

```

Looks like you are in need of an upgrade. By the way, julian time  
explicitly includes no leap seconds, meaning you don't have to think  
about them. In this sense, it is not tied to any physical period (just  
as seconds since Jan 1, 1970 is not) -- it's simply a continuous time  
base for ease of intercomparison.

See:

[http://hpiers.obspm.fr/webiers/general/syframes/convent/UAI\\_b1.html](http://hpiers.obspm.fr/webiers/general/syframes/convent/UAI_b1.html)

(credit to Tom McGlynn for the link)

