Subject: Re: Faster way to convert a string array of dates to julian dates Posted by henrygroe on Mon, 24 Sep 2007 15:10:30 GMT

View Forum Message <> Reply to Message

```
On Sep 24, 7:49 am, "Ryan." <rchug...@gmail.com> wrote:
> Hi All,
>
> I want to convert an string array of dates to julian numbers in the
> most efficient way in IDL. The only way I can come up with doing it
> is using a FOR loop but I'm hoping that some of you could help me to
> utilize IDL's super-duper array manipulation strengths. This is what
> I have:
>
> date format: YYYY-MM-DD HH:MM:SS.FFFF+TT
> F->fractions of a second
> T->time zone specification (e.g. -05 for Eastern Standard Time, etc.)
>
> I don't care much about the time zone because all the dates are given
> in UTC. It doesn't need to be included in the conversion.
Siven a string array like this:> dates = ['2007-09-21 12:15:00.0000+00', '2007-09-22]
23:25:15.9999+00', ...]
> I can convert it using this code:
> juls = DBLARR(N_ELEMENTS(dates))
> FOR i=0, N_ELEMENTS(dates)-1 DO BEGIN
> splitdate = STRSPLIT(dates[i], ': +-', /EXTRACT)
> juls[i] = JULDAY(splitdate[1], splitdate[2], splitdate[0],
> splitdate[3], splitdate[4], splitdate[5])
> ENDFOR
>
> This works, but would like to know a more efficient way of converting
> it because I do this quite often.
>
> Thanks.
> Ryan.
If the format of your input strings are never changing (which you're
pretty much implicitly assuming with strsplit anyway), then:
juls = julday(strmid(dates,0,4),strmid(dates,5,2),strmid(dates,
8,2),strmid(dates,11,2),strmid(dates,14,2),strmid(dates,17,7))
```

Subject: Re: Faster way to convert a string array of dates to julian dates Posted by Pedro.Rodriguez.Pascu on Tue, 25 Sep 2007 14:09:23 GMT View Forum Message <> Reply to Message

```
On 24 sep, 14:49, "Ryan." <rchug...@gmail.com> wrote:
> Hi All.
>
> I want to convert an string array of dates to julian numbers in the
> most efficient way in IDL. The only way I can come up with doing it
> is using a FOR loop but I'm hoping that some of you could help me to
> utilize IDL's super-duper array manipulation strengths. This is what
> I have:
> date format: YYYY-MM-DD HH:MM:SS.FFFF+TT
> F->fractions of a second
> T->time zone specification (e.g. -05 for Eastern Standard Time, etc.)
>
> I don't care much about the time zone because all the dates are given
> in UTC. It doesn't need to be included in the conversion.
Siven a string array like this:> dates = ['2007-09-21 12:15:00.0000+00', '2007-09-22
23:25:15.9999+00', ...]
> I can convert it using this code:
> juls = DBLARR(N ELEMENTS(dates))
> FOR i=0, N_ELEMENTS(dates)-1 DO BEGIN
> splitdate = STRSPLIT(dates[i], ': +-', /EXTRACT)
> juls[i] = JULDAY(splitdate[1], splitdate[2], splitdate[0],
> splitdate[3], splitdate[4], splitdate[5])
> ENDFOR
> This works, but would like to know a more efficient way of converting
> it because I do this quite often.
> Thanks,
> Ryan.
Hello,
you may want to explore the possibilities of c() format.
Unfortunately, as far as I know, it cannot deal with time zones yet.
The following lines would do the job avoiding FOR loops:
fmt='(C(CYI4,1X,CMOI2,1X,CDI2,1X,CHI2,1X,CMI2,1X,CSF7.4))'
iuls=DBLARR(N ELEMENTS(dates))
READS, dates, juls, FORMAT=fmt
tz=LONG(STRMID(dates,24))
juls=juls-tz/24.
```

Subject: Re: Faster way to convert a string array of dates to julian dates Posted by henrygroe on Tue, 25 Sep 2007 18:31:57 GMT

```
View Forum Message <> Reply to Message
```

```
On Sep 24, 8:10 am, henryg...@gmail.com wrote:
> On Sep 24, 7:49 am, "Ryan." <rchug...@gmail.com> wrote:
>
>
>
>> Hi All,
>> I want to convert an string array of dates to julian numbers in the
>> most efficient way in IDL. The only way I can come up with doing it
>> is using a FOR loop but I'm hoping that some of you could help me to
>> utilize IDL's super-duper array manipulation strengths. This is what
>> I have:
>> date format: YYYY-MM-DD HH:MM:SS.FFFF+TT
>> F->fractions of a second
>> T->time zone specification (e.g. -05 for Eastern Standard Time, etc.)
>
>> I don't care much about the time zone because all the dates are given
>> in UTC. It doesn't need to be included in the conversion.
>> Given a string array like this:> dates = ['2007-09-21 12:15:00.0000+00', '2007-09-22
23:25:15.9999+00', ...]
>> I can convert it using this code:
>> iuls = DBLARR(N_ELEMENTS(dates))
>> FOR i=0, N ELEMENTS(dates)-1 DO BEGIN
>> splitdate = STRSPLIT(dates[i], ': +-', /EXTRACT)
>> juls[i] = JULDAY(splitdate[1], splitdate[2], splitdate[0],
>> splitdate[3], splitdate[4], splitdate[5])
>> ENDFOR
>> This works, but would like to know a more efficient way of converting
>> it because I do this quite often.
>
>> Thanks,
>> Ryan.
>
> If the format of your input strings are never changing (which you're
> pretty much implicitly assuming with strsplit anyway), then:
> juls = julday(strmid(dates,0,4),strmid(dates,5,2),strmid(dates,
```

> 8,2),strmid(dates,11,2),strmid(dates,14,2),strmid(dates,17,7))

WHOOPS! I didn't read the IDL help entry for julday closely enough. For some reason they do Month, Day, Year, Hour, Minute, Second. (Which makes little sense to me. I just assumed it was Year, Month, Day....)