Subject: Re: JULDAY 5.4 not same as 5.3?
Posted by Mark Hadfield on Thu, 01 Mar 2001 23:19:15 GMT
View Forum Message <> Reply to Message

> "Don Woodraska" <don.woodraska@lasp.colorado.edu> wrote in message news:3A9ED47E.F92EF914@lasp.colorado.edu...
> Has anyone else noticed a bug in JULDAY that appeared in 5.4?
> IDL> help,umm,udd,uyear,uhr,umin,usec
> UMM LONG = 2
> UDD LONG = 16
> UYEAR LONG = 2001
> UHR ULONG = 0
> UMIN ULONG = 0
> USEC ULONG = 0
> IDL> in_jday = julday(umm, udd, uyear, uhr, umin, usec)
> IDL> help,in_jday
> IN_JDAY DOUBLE = 1.8140893e+08

I get the right answer under 5.4 (win32 x86):

IDL> help,umm,udd,uyear,uhr,umin,usec
UMM LONG = 2
UDD LONG = 16
UYEAR LONG = 2001
UHR LONG = 0
UMIN LONG = 0
USEC LONG = 0
IDL> print, julday(umm, udd, uyear, uhr, umin, usec)
2451956.5

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield
National Institute for Water and Atmospheric Research

Subject: Re: JULDAY 5.4 not same as 5.3?
Posted by Don Woodraska on Fri, 02 Mar 2001 00:28:25 GMT
View Forum Message <> Reply to Message

I should have included this in the original post:

IDL> print,!version
{ sparc sunos unix 5.4 Sep 25 2000 64 64}

Thanks for your response Mark. I'll bet it must be a 64-bit-only bug.
Can anyone else verify this bug on another 64-bit machine?

Interestingly, if I don't pass the hour, minute, and second, then I get
the correct answer with version 5.4.

Thanks,
Don

Mark Hadfield wrote:

>> "Don Woodraska" <don.woodraska@lasp.colorado.edu> wrote in message
> news:3A9ED47E.F92EF914@lasp.colorado.edu...
>> Has anyone else noticed a bug in JULDAY that appeared in 5.4?
>> IDL> help,umm,udd,uyear,uhr,umin,usec
>> UMM LONG = 2
>> UDD LONG = 16
>> UYEAR LONG = 2001
>> UHR ULONG = 0
>> UMIN ULONG = 0
>> USEC ULONG = 0
>> IDL> in_jday = julday(umm, udd, uyear, uhr, umin, usec)
>> IDL> help,in_jday
>> IN_JDAY DOUBLE = 1.8140893e+08
>
> I get the right answer under 5.4 (win32 x86):
>
> IDL> help,umm,udd,uyear,uhr,umin,usec
> UMM LONG = 2
> UDD LONG = 16
> UYEAR LONG = 2001
> UHR LONG = 0
> UMIN LONG = 0
> USEC LONG = 0
> IDL> print, julday(umm, udd, uyear, uhr, umin, usec)
> 2451956.5
>
> ---
> Mark Hadfield
> m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield
> National Institute for Water and Atmospheric Research

Subject: Re: JULDAY 5.4 not same as 5.3?
Posted by pertti on Fri, 02 Mar 2001 10:29:55 GMT
View Forum Message <> Reply to Message

Don Woodraska (don.woodraska@lasp.colorado.edu) wrote:
: I should have included this in the original post:
:
: IDL> print,!version

: { sparc sunos unix 5.4 Sep 25 2000 64 64}
:
: Thanks for your response Mark. I'll bet it must be a 64-bit-only bug.
: Can anyone else verify this bug on another 64-bit machine?

Mhhh...

IDL> umm=2l & udd=16l & uyear=2001l
IDL> uhr=0l & umin=0l & usec=0l
IDL> in_jday = julday(umm, udd, uyear, uhr, umin, usec)
IDL> help,in_jday
IN_JDAY DOUBLE = 2451956.5
IDL> print,!version
{ sparc sunos unix 5.4 Sep 25 2000 32 64}
IDL> $uname -a
SunOS sun4 5.8 Generic_108528-03 sun4u sparc SUNW,Ultra-Enterprise
IDL> $/bin/isainfo -b
64

According to IDL Online Help, there are two versions of IDL 5.4 for Solaris, 64-bit and 32-bit memory support. We have 64-bit kernel (Solaris 8), but still seem to use 32-bit memory support (if I understand correctly the contents of !version system variable). If you have both 64-bit and 32-bit versions installed, it should be possible to start IDl in 32-bit mode by giving option -32.

Pertti

Subject: Re: JULDAY 5.4 not same as 5.3?
Posted by Don Woodraska on Fri, 02 Mar 2001 16:49:16 GMT
View Forum Message <> Reply to Message

Thanks for the tip. I launched IDL with the -32 option, but it still doesn't work right. JULDAY behaves incorrectly in certain cases which I will describe below. Unfortunately Pertti, you used long argument and not ulong arguments, which I suspect, is why it worked for you.

SUMMARY:
***
The bug appears only in unsigned long and unsigned 64-bit long when all 6 args are present, and appearantly does not affect Windows versions.

Fix: typecast all 6 args to be long when calling JULDAY.
***

DETAILS:

This bug is independent of whether or not I'm running in 64-bit or 32-bit memory mode. Since everything before this was in 64-bit mode, I'll show you everything here in 32-bit mode.

```
IDL> print,!version
{ sparc sunos unix 5.4 Sep 25 2000      32      64}
IDL> $uname -a
SunOS sun4 5.7 Generic_106541-12 sun4u sparc SUNW,Ultra-30
IDL> mo=long(2) & day=long(16) & year=long(2001) & h=long(0) & m=long(0) &
s=long(0)
IDL> x=julday(mo,day,year,h,m,s)
IDL> help,x
X               DOUBLE    =       2451956.5
```

As expected, it works fine with all 6 long arguments, however, with ulong...

```
IDL> mo=ulong(2) & day=ulong(16) & year=ulong(2001) & h=ulong(0) &
m=ulong(0) & s=ulong(0)
IDL> x=julday(mo,day,year,h,m,s)
IDL> help,x
X               DOUBLE    =    1.8140893e+08
```

This is not correct, however, if we leave off the h,m,s args...

```
IDL> mo=ulong(2) & day=ulong(16) & year=ulong(2001)
IDL> x=julday(mo,day,year)
IDL> help,x
X               LONG      =      2451957
```

It works! (This is the new Julian day starting at noon on Feb 16,2001.)

To demonstrate the ulong64 problem, here's more sample output.

```
IDL> mo=ulong64(2) & day=ulong64(16) & year=ulong64(2001) & h=ulong64(0) &
m=ulong64(0) & s=ulong64(0)
IDL> x=julday(mo,day,year,h,m,s)
IDL> help,x
X               DOUBLE    =    7.6861434e+17
IDL> x=julday(mo,day,year)
IDL> help,x
X               LONG      =      2451957
```

I think we've made some progress in understanding the bug. Incorrect results are reported when 6 unsigned-long or unsigned-64-bit-long parameters are passed to JULDAY. Unsigned integer, integer, long, and long64 all seem to work properly.

The problem seems to be isolated to ULONG and ULONG64 independent of whether running in 32-bit or 64-bit memory mode. Passing ulong and ulong64 parameters each give different wrong answers.

I'm getting around this by typecasting all my arguments to be long in the call to JULDAY.

We still have the IDL lib directory for the old 5.3 versions. The old procedure works even under the new 64-bit version regardless of input parameter data type. This leads me to believe that the bug is actually hidden in the JULDAY.PRO file in the IDL lib directory. Look at the differences in code between the two versions. It is drastic.

Lesson learned: typecast arguments to JULDAY to be long.

Cheers,
Don

Pertti Rautiainen wrote:

> Don Woodraska (don.woodraska@lasp.colorado.edu) wrote:
> : I should have included this in the original post:
> :
> : IDL> print,!version
> : { sparc sunos unix 5.4 Sep 25 2000 64 64}
> :
> : Thanks for your response Mark. I'll bet it must be a 64-bit-only bug.
> : Can anyone else verify this bug on another 64-bit machine?
>
> Mhhh...
>
> IDL> umm=2l & udd=16l & uyear=2001l
> IDL> uhr=0l & umin=0l & usec=0l
> IDL> in_jday = julday(umm, udd, uyear, uhr, umin, usec)
> IDL> help,in_jday
> IN_JDAY DOUBLE = 2451956.5
> IDL> print,!version
> { sparc sunos unix 5.4 Sep 25 2000 32 64}
> IDL> $uname -a
> SunOS sun4 5.8 Generic_108528-03 sun4u sparc SUNW,Ultra-Enterprise
> IDL> $/bin/isainfo -b
> 64
>
> According to IDL Online Help, there are two versions of IDL 5.4 for
> Solaris, 64-bit and 32-bit memory support. We have 64-bit
> kernel (Solaris 8), but still seem to use 32-bit memory support (if I
> understand correctly the contents of !version system variable). If you

> have both 64-bit and 32-bit versions installed, it should be possible
> to start IDl in 32-bit mode by giving option -32.
>
> Pertti

--
Donald Woodraska, Phone: 303-735-5617, Fax: 303-735-4843
Laboratory for Atmospheric and Space Physics, University of Colorado
Campus Box 590, 1234 Innovation Drive, Boulder, CO 80303

Subject: Re: JULDAY 5.4 not same as 5.3?
Posted by Chris Torrence on Fri, 02 Mar 2001 16:52:00 GMT
View Forum Message <> Reply to Message

Hi Don,

The problem is with using an unsigned long integer. Inside JULDAY.PRO,
it subtracts 12 off of the hours before dividing by 24. This will cause
wrap-around for unsigned longs and unsigned long64s.

So you should be able to fix (no pun!) your problem by:

```
in_jday = julday( umm, udd, uyear, FIX(uhr), umin, usec )
```

This bug has been logged with RSI, and will be fixed in the next IDL
version.

Cheers,
Chris Torrence
Research Systems, Inc.

Don Woodraska wrote:
>
> Has anyone else noticed a bug in JULDAY that appeared in 5.4?
>
> I tried this in IDL 5.3:
>
> IDL> help,umm,udd,uyear,uhr,umin,usec
> UMM LONG = 2
> UDD LONG = 16
> UYEAR LONG = 2001
> UHR ULONG = 0
> UMIN ULONG = 0
> USEC ULONG = 0
> IDL> in_jday = julday(umm, udd, uyear, uhr, umin, usec)
> IDL> gps0_jday = julday(1,6,1980,0,0,0)

>
...
> IDL> in_jday = julday(umm, udd, uyear, uhr, umin, usec)
> IDL> gps0_jday = julday(1,6,1980,0,0,0)
> IDL> help,in_jday,gps0_jday
> IN_JDAY DOUBLE = 1.8140893e+08
> GPS0_JDAY DOUBLE = 2444244.5
>

Subject: Re: JULDAY 5.4 not same as 5.3?
Posted by Don Woodraska on Fri, 02 Mar 2001 18:39:21 GMT
View Forum Message <> Reply to Message

I've pinpointed the problem.

SUMMARY:
***
The bug appears only with unsigned-long and unsigned-64-bit-long hour argument

to JULDAY.

The bug appears on lines 178-179 of JULDAY.PRO. Here it is:
```
 jul = TEMPORARY(JUL) + ( (TEMPORARY(d_Hour)-12)/24d + $
  TEMPORARY(d_Minute)/1440d + TEMPORARY(d_Second)/86400d + eps )
```

FIX:
***
Here's a fix:
```
 jul = TEMPORARY(JUL) + ( (TEMPORARY(double(d_Hour))-12)/24d + $
  TEMPORARY(d_Minute)/1440d + TEMPORARY(d_Second)/86400d + eps )
```

WHY IT WORKS:
***
When you take the difference of 2 different data types (dhour and long 12) the

result is automatically promoted to the data type with the most precision (at least on our unix implementation).

Although an unsigned long has no more bits/higher precision than a long, it is

first in the expression evaluation. The result of 0ul-12l is an unsigned long (4294967284 if you want to put a value to it). If you reverse the order like this,
-12l+0ul, you get a long.

When a ulong64 is passed as the hour argument we get the expression

ulong64(0)-12l, which gives 18446744073709551604.

Cheers,
Don
