

---

Subject: Re: formatting output from caldat to label animation frames

Posted by [btupper](#) on Thu, 03 Oct 2002 10:20:43 GMT

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

---

On 2 Oct 2002 23:27:16 -0700, [twright@usgs.gov](mailto:twright@usgs.gov) (Thomas Wright) wrote:

> I have been working with Julian dates and would like to label  
> animation frames with date and time expressed as mm/dd/yyyy  
> hh:mm:ss. I can get the values for month, day, year, hour, minute and  
> second using CALDAT, but they are widely spaced, and when I add '/' or  
> ':' I get something like:

>  
> 7/ 26/ 1981 6: 17: 56  
>

> instead of 7/26/1981 6:17:56.

> How can I create variables without spaces from CALDAT output to get  
> normal looking text?

Hi,

I think you are seeing the effects of converting integers to string  
without using STRTRIM() to remove leading spaces from each integer.  
You could also use the format keyword in a call to string to remove  
blanks. The leading blanks arise from the conversion of an n byte  
number (like long integers are 4 bytes) where some of the bytes are  
acting as place holders.

```
IDL> print, 'this is an unstripped:', String(17L), ' number'
this is an unstripped:      17 number
IDL> print, 'this is an stripped:', StrTrim(17L,2), ' number'
this is an stripped:17 number
IDL> print, 'this is an stripped:', String(17L,format = '(I0)'), '
number'
this is an stripped:17 number
IDL> print, 'this is an stripped:', String(17L,format = '(I7)'), '
number'
this is an stripped:   17 number
```

I have added below a routine I use for formating time stamps just as  
you describe.

Ben

```
;+
; NAME:
; Jul2Cal
;
```

```

; PURPOSE:
; This function is a wrapper around the CalDat procedure. Use
this function
; to convert a Julian Day Number to a double precision array of
; [year, month, day, hour, minute, second].
; A string date/time can be optionally returned.
;
; CATEGORY:
; Date/Time
;
; CALLING SEQUENCE:
; result = Jul2Cal([julian], [/TO_STRING], [/MDY])
;
; ARGUMENTS:
; Julian (optional) A scalar or array of Julian Day Numbers. If
not provided
; the current time is used.
;
; KEYWORDS:
; TO_STRING Set this keyword to return an array of date/time
strings.
; The first column is the date in the YYYY-MM-DD format.
; The second column is the time in the HH:MM:SS format.
; Fractions of seconds are not preserved.
;
; MDY Set this keyword to change the string order of the Date to
MM-DD-YYY.
;
; DATE_SEPARATOR Set this keyword to the date separator. The
default is '-'
;
; TIME_SEPARATOR Set this keyword to the time separator. The
default is ':'
;
; EXAMPLE:
;IDL> Print, Jul2Cal(), Jul2Cal(/To_String), Jul2Cal(/To_String, /MDY)
; 2001.0000 11.000000 27.000000 15.000000
41.000000 9.0000460
;2001-11-27 15:41:09
;11-27-2001 15:41:09
;
; MODIFICATION HISTORY:
; 27 NOV 2001 Written by Ben Tupper.
; pemaquidriver@tidewater.net
; 1 MAR 2002 Added Time and Date separator controls. BT
;-

```

```

FUNCTION Jul2Cal, julian, To_String = To_String, MDY = MDY, $

```

```
Date_Separator = Date_Separator, Time_Separator =  
Time_Separator
```

```
If n_elements(Julian) EQ 0 Then Julian = SysTime(/Julian)  
CalDat, Julian, Month, Day, Year, Hour, Minute, Second
```

```
If Keyword_Set(To_String) Then Begin
```

```
  If n_elements(Date_Separator) EQ 0 Then DS = '-' Else DS =  
  Date_Separator[0]
```

```
  If n_elements(Time_Separator) EQ 0 Then TS = ':' Else TS =  
  Time_Separator[0]
```

```
  Sec = StrTrim(Round(Second),2)
```

```
  Len = StrLen(Sec)
```

```
  A = Where(Len LT 2, cnt)
```

```
  If cnt GT 0 Then Sec[A] = '0' + Sec[A]
```

```
  Mins = StrTrim(Minute,2)
```

```
  Len = StrLen(Mins)
```

```
  A = Where(Len LT 2, cnt)
```

```
  If cnt GT 0 Then Mins[A] = '0' + Mins[A]
```

```
  Hrs = StrTrim(Hour,2)
```

```
  Len = StrLen(Hrs)
```

```
  A = Where(Len LT 2, cnt)
```

```
  If cnt GT 0 Then Hrs[A] = '0' + Hrs[A]
```

```
  Dy = StrTrim(Day,2)
```

```
  Len = StrLen(Dy)
```

```
  A = Where(Len LT 2, cnt)
```

```
  If cnt GT 0 Then Dy[A] = '0' + Dy[A]
```

```
  Mnth = StrTrim(Month,2)
```

```
  Len = StrLen(Mnth)
```

```
  A = Where(Len LT 2, cnt)
```

```
  If cnt GT 0 Then Mnth[A] = '0' + Mnth[A]
```

```
  If Keyword_Set(MDY) Then Date = Mnth + DS + Dy+ DS +  
  StrTrim(Year,2) Else $
```

```
  Date = StrTrim(Year,2) + DS + Mnth + DS + Dy
```

```
  Return, Transpose([[Date], [Hrs+TS+Mins+TS+Sec]])
```

```
EndIf Else Return, Transpose([[Year], [Month], [Day], [Hour],  
[Minute], [Second]])
```

```
END
```

---