

[Home](#) | [Categories](#) | [Alphabetical](#) | [Classes](#) | [All Contents](#) | [\[< \]](#) | [\[> \]](#)

CALDAT

[Syntax](#) | [Arguments](#) | [Keywords](#) | [Examples](#) | [Version History](#) | [See Also](#)

The CALDAT procedure computes the month, day, year, hour, minute, or second corresponding to a given Julian date. The inverse of this procedure is JULDAY.

Note

The Julian calendar, established by Julius Caesar in the year 45 BCE, was corrected by Pope Gregory XIII in 1582, excising ten days from the calendar. The CALDAT procedure reflects the adjustment for dates after October 4, 1582. See the example below for an illustration.

This routine is written in the IDL language. Its source code can be found in the file `caldat.pro` in the `lib` subdirectory of the IDL distribution.

Syntax

CALDAT, *Julian*, *Month* [, *Day* [, *Year* [, *Hour* [, *Minute* [, *Second*]]]]]

Arguments

Julian

A numeric value or array that specifies the Julian Day Number (which begins at noon) to be converted to a calendar date.

Note

Julian values must be in the range -1095 to 1827933925, which corresponds to calendar dates 1 Jan 4716 B.C.E. and 31 Dec 5000000, respectively.

Note

Julian Day Numbers should be maintained as double-precision floating-point data when the numbers are used to determine hours, minutes, and seconds.

Month

A named variable that, on output, contains a longword integer or longword integer array representing the number of the desired month (1 = January, ..., 12 = December).

Day

A named variable that, on output, contains a longword integer or longword integer array representing the number of the day of the month (1-31).

Year

A named variable that, on output, contains a longword integer or longword integer array representing the number of the desired year (e.g., 1994).

Hour

A named variable that, on output, contains a longword integer or longword integer array representing the number of the hour of the day (0-23).

Minute

A named variable that, on output, contains a longword integer or longword integer array representing the number of the minute of the hour (0-59).

Second

A named variable that, on output, contains a double-precision floating-point value or a double-precision floating-point array representing the number of the second of the minute (0-59).

Keywords

None.

Examples

In 1582, Pope Gregory XIII adjusted the Julian calendar to correct for its inaccuracy of slightly more than 11 minutes per year. As a result, the day following October 4, 1582 was October 15, 1582. CALDAT follows this convention, as illustrated by the following commands:

```
CALDAT, 2299160, Month1, Day1, Year1
CALDAT, 2299161, Month2, Day2, Year2
PRINT, Month1, Day1, Year1
PRINT, Month2, Day2, Year2
```

IDL prints:

```
10      4      1582
10     15      1582
```

Warning

You should be aware of this discrepancy between the original and revised Julian calendar reckonings if you calculate dates before October 15, 1582.

Be sure to distinguish between *Month* and *Minute* when assigning variable names. For example, the following code would cause the Month value to be the same as the Minute value:

```
;Find date corresponding to Julian day 2529161.36:
CALDAT, 2529161.36, M, D, Y, H, M, S
PRINT, M, D, Y, H, M, S
```

IDL prints:

```
0          4          2212          18          0          0.00000000
```

Moreover, Julian Day Numbers should be maintained as double-precision floating-point data when the numbers are used to determine hours, minutes, and seconds.

So, instead of the previous call to CALDAT, use something like:

```
CALDAT, 2529161.36D, Month, Day, Year, Hour, Minute, Second
PRINT, Month, Day, Year, Hour, Minute, Second
```

IDL prints:

```
7          4          2212          20          38          23.999989
```

You can also use arrays for the *Julian* argument:

```
CALDAT, DINDGEN(4) + 2449587.0D, m, d, y  
PRINT, m, d, y
```

IDL prints:

8	8	8	8
22	23	24	25
1994	1994	1994	1994

Version History

Introduced: Pre 4.0

See Also

[BIN_DATE](#), [JULDAY](#), [SYSTIME](#)

[Home](#) | [Categories](#) | [Alphabetical](#) | [Classes](#) | [All Contents](#) | [\[< \]](#) | [\[> \]](#)