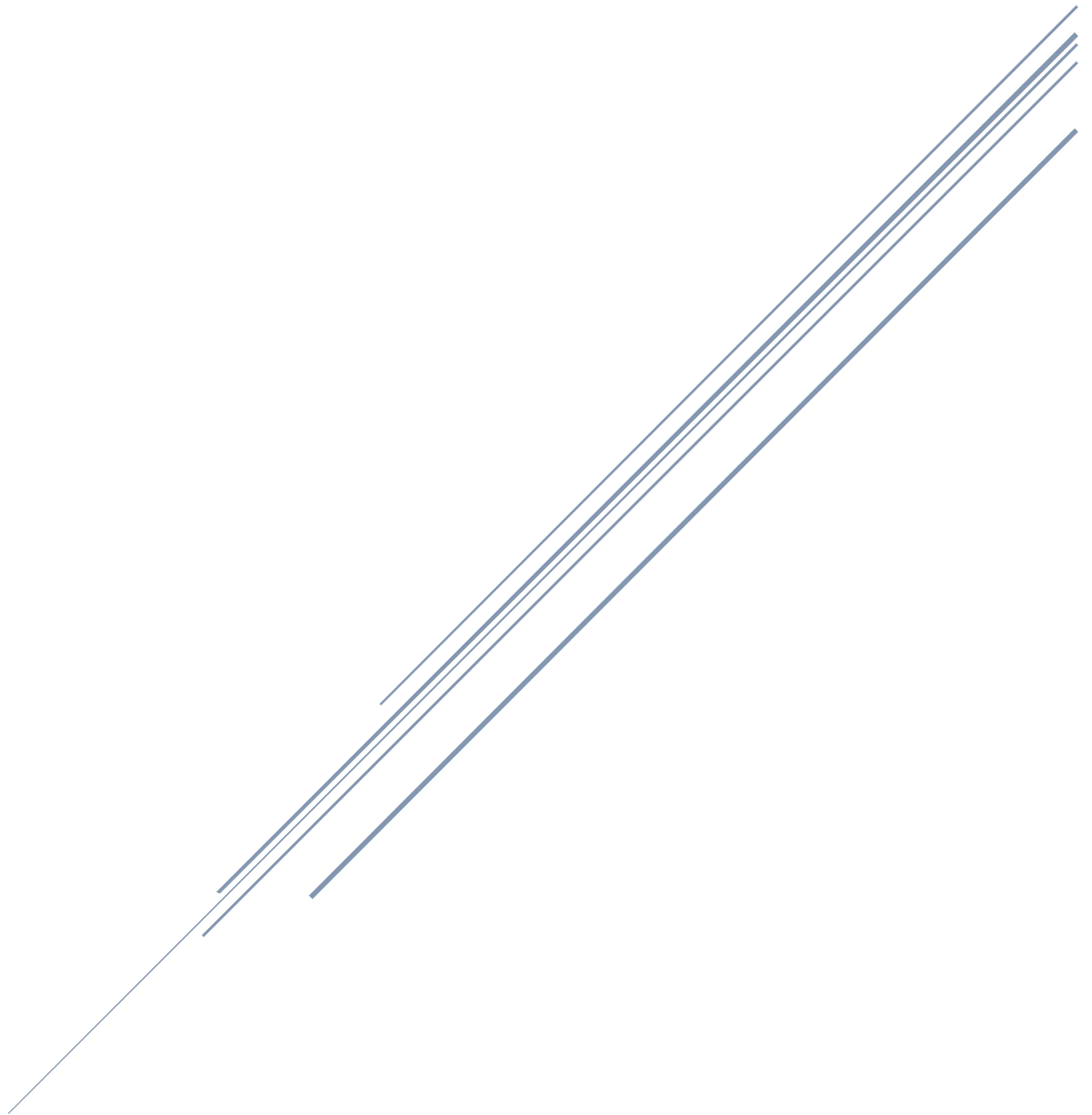


Splunk

Date and time format variables



Date and time format variables

The variables that can be used to specify time formats in the evaluation methods **strftime()** and **strptime()** are listed in this article. These variables can also be used to describe timestamps in event data.

Date and time variables

Variable	Description
%c	The server's operating system's definition of the date and time in the current locale. As an illustration, Thu Jul 18 09:30:00 2019 for Linux and US English.
%+	The time and date in local time zone format as specified by the server's operating system. For US English on Linux, for example, Thu Jul 18 09:30:00 PDT 2019 .

Splunk Time Variables List (Alphabetical)

Time variables

Description:

Splunk provides various time variables that allow you to manipulate and analyze timestamps in your data. Here is an alphabetical list of commonly used time variables in Splunk:

%A	Full weekday name (e.g., Wednesday)
%a	Abbreviated weekday name (e.g., Wed)
%B	Full month name (e.g., June)
%b	Abbreviated month name (e.g., Jun)
%c	Locale's appropriate date and time representation (e.g., Wed Jun 2 10 30 45 2023)
%D	Equivalent to %m/%d/%y (e.g., 06/02/23)
%d	Day of the month as a zero-padded decimal number (e.g., 02)
%e	Day of the month as a decimal number (e.g., 2)
%H	Hour (24-hour clock) as a zero-padded decimal number (e.g., 10)
%I	Hour (12-hour clock) as a zero-padded decimal number (e.g., 02)
%j	Day of the year as a zero-padded decimal number (e.g., 153)
%k	Hour (24-hour clock) as a decimal number (e.g., 10)
%l	Hour (12-hour clock) as a decimal number (e.g., 2)
%M	Minute as a zero-padded decimal number (e.g., 30)
%N	Nanoseconds (000000000-999999999)
%n	Newline character
%P	Locale's equivalent of either AM or PM (e.g., am)
%p	Locale's equivalent of either AM or PM (e.g., AM)
%R	Equivalent to %H %M (e.g., 10 30)
%r	Locale's 12-hour clock time (e.g., 10 30 45)

%S	Second as a zero-padded decimal number (e.g., 45)
%s	Unix timestamp (number of seconds since the Unix epoch, January 1, 1970)
%T	Equivalent to %H %M %S (e.g., 10 30 45)
%t	Tab character
%U	Week number of the year (Sunday as the first day of the week) as a zero-padded decimal number (e.g., 22)
%u	Weekday as a decimal number (1-7, where Monday is 1)
%V	ISO 8601 week number of the year (Monday as the first day of the week) as a zero-padded decimal number (e.g., 22)
%v	Equivalent to %e-%b-%Y (e.g., 2-Jun-2023)
%W	Week number of the year (Monday as the first day of the week) as a zero-padded decimal number (e.g., 22)
%w	Weekday as a decimal number (0-6, where Sunday is 0)
%x	Locale's appropriate date representation (e.g., 06/02/23)
%X	Locale's appropriate time representation (e.g., 10 30 45)
%Y	Year with century as a decimal number (e.g., 2023)
%y	Year without century as a decimal number (e.g., 23)
%Z	Time zone name (e.g., PDT)
%z	UTC offset in the form +HHMM or -HHMM (e.g., +0530)

Date variables

%F	Equivalent to %Y-%m-%d (the ISO 8601 date format).
%x	The date in the format of the current locale. For example, 7/13/2019 for US English.

Specifying days and weeks

Variable	Description
%A	Full weekday name. (Sunday, ..., Saturday)
%a	Abbreviated weekday name. (Sun, ..., Sat)
%d	Day of the month as a decimal number, includes a leading zero. (01 to 31)
%e	Like %d, the day of the month as a decimal number, but a leading zero is replaced by a space. (1 to 31)
%j	Day of year as a decimal number, includes a leading zero. (001 to 366)
%V (or %U)	Week of the year. The %V variable starts the count at 1, which is the most common start number. The %U variable starts the count at 0.
%w	Weekday as a decimal number. (0 = Sunday, ..., 6 = Saturday)

Example:

index=your_index

```
| eval weekday_full = strftime(_time, "%A"),  
      weekday_abbrev = strftime(_time, "%a"),  
      day_zero_padded = strftime(_time, "%d"),  
      day_no_padding = strftime(_time, "%e"),  
      day_of_year = strftime(_time, "%j"),  
      iso_week_number = strftime(_time, "%V"),  
      week_number = strftime(_time, "%U"),  
      weekday_decimal = strftime(_time, "%w")  
| table _time, weekday_full, weekday_abbrev, day_zero_padded, day_no_padding,  
      day_of_year, iso_week_number, week_number, weekday_decimal
```

Output:

time	weekday_full	weekday_abbrev	day_zero_padded	day_no_padding	day_of_year	iso_week_number	week_number	weekday_decimal
2023-06-02 10:30:45	Friday	Fri	02	2	153	22	22	5
2023-06-03 15:45:20	Saturday	Sat	03	3	154	22	22	6
2023-06-04 08:20:10	Sunday	Sun	04	4	155	22	23	0
2023-06-05 18:55:30	Monday	Mon	05	5	156	23	23	1

Specifying months

Variable	Description
%b	Abbreviated month name. (Jan, Feb, etc.)
%B	Full month name. (January, February, etc.)
%m	Month as a decimal number. (01 to 12). Leading zeros are accepted but not required.

Example:

index=your_index

```
| eval month_abbrev = strftime(_time, "%b"),  
      month_full = strftime(_time, "%B"),  
      month_zero_padded = strftime(_time, "%m")  
| table _time, month_abbrev, month_full, month_zero_padded
```

Output:

_time	month_abbrev	month_full	month_zero_padded
2023-06-02 10:30:45	Jun	June	06
2023-07-15 08:45:20	Jul	July	07
2023-08-28 14:20:10	Aug	August	08
2023-09-10 18:55:30	Sep	September	09

Specifying year

Variable	Description
%y	Year as a decimal number, without the century. (00 to 99). Leading zeros are accepted but not required.
%Y	Year as a decimal number with century. For example, 2020.

Example:

```
index=your_index
| eval year_without_century = strftime(_time, "%y"),
    year_with_century = strftime(_time, "%Y")
| table _time, year_without_century, year_with_century
```

Output:

_time	year_without_century	year_with_century
2023-06-02 10:30:45	23	2023
2024-07-15 08:45:20	24	2024
2025-08-28 14:20:10	25	2025
2026-09-10 18:55:30	26	2026