NAME

systemd.time − Time and date specifications

DESCRIPTION

In systemd, timestamps, time spans, and calendar events are displayed and may be specified in closely related syntaxes.

DISPLAYING TIME SPANS

Time spans refer to time durations. On display, systemd will present time spans as a space−separated series of time values each suffixed by a time unit. Example:

2h 30min

All specified time values are meant to be added up. The above hence refers to 150 minutes. Display is locale−independent, only English names for the time units are used.

PARSING TIME SPANS

When parsing, systemd will accept the same time span syntax. Separating spaces may be omitted. The following time units are understood:

- usec, us, µs
- msec, ms
- seconds, second, sec, s
- minutes, minute, min, m
- hours, hour, hr, h
- days, day, d
- weeks, week, w
- months, month, M (defined as 30.44 days)
- years, year, y (defined as 365.25 days)

If no time unit is specified, generally seconds are assumed, but some exceptions exist and are marked as such. In a few cases "ns", "nsec" is accepted too, where the granularity of the time span permits this. Parsing is generally locale−independent, non−English names for the time units are not accepted.

Examples for valid time span specifications:

2 h
2hours
48hr
1y 12month
55s500ms
300ms20s 5day

One can use the **timespan** command of **systemd-analyze**(1) to normalise a textual time span for testing and validation purposes.

DISPLAYING TIMESTAMPS

Timestamps refer to specific, unique points in time. On display, systemd will format these in the local timezone as follows:

Fri 2012−11−23 23:02:15 CET

The weekday is printed in the abbreviated English language form. The formatting is locale−independent.

In some cases timestamps are shown in the UTC timezone instead of the local timezone, which is indicated via the "UTC" timezone specifier in the output.

In some cases timestamps are shown with microsecond granularity. In this case the sub−second remainder

is separated by a full stop from the seconds component.

PARSING TIMESTAMPS

When parsing, systemd will accept a similar syntax, but expects no timezone specification, unless it is given as the literal string "UTC" (for the UTC timezone), or is specified to be the locally configured timezone, or the timezone name in the IANA timezone database format. The complete list of timezones supported on your system can be obtained using the "timedatectl list−timezones" (see **timedatectl**(1)). Using IANA format is recommended over local timezone names, as less prone to errors (eg: with local timezone it's possible to specify daylight saving time in winter, while it's incorrect). The weekday specification is optional, but when the weekday is specified, it must either be in the abbreviated ("Wed") or non−abbreviated ("Wednesday") English language form (case does not matter), and is not subject to the locale choice of the user. Either the date, or the time part may be omitted, in which case the current date or 00:00:00, respectively, is assumed. The seconds component of the time may also be omitted, in which case ":00" is assumed. Year numbers may be specified in full or may be abbreviated (omitting the century).

A timestamp is considered invalid if a weekday is specified and the date does not match the specified day of the week.

When parsing, systemd will also accept a few special placeholders instead of timestamps: "now" may be used to refer to the current time (or of the invocation of the command that is currently executed). "today", "yesterday", and "tomorrow" refer to 00:00:00 of the current day, the day before, or the next day, respectively.

When parsing, systemd will also accept relative time specifications. A time span (see above) that is prefixed with "+" is evaluated to the current time plus the specified time span. Correspondingly, a time span that is prefixed with "−" is evaluated to the current time minus the specified time span. Instead of prefixing the time span with "+" or "−", it may also be suffixed with a space and the word "left" or "ago".

Finally, a timespan prefixed with "@" is evaluated relative to the UNIX time epoch 1st Jan, 1970, 00:00.

Examples for valid timestamps and their normalized form (assuming the current time was 2012−11−23 18:15:22 and the timezone was UTC+8, for example TZ=Asia/Shanghai):

```
 Fri 2012−11−23 11:12:13 → Fri 2012−11−23 11:12:13
    2012−11−23 11:12:13 → Fri 2012−11−23 11:12:13
2012−11−23 11:12:13 UTC → Fri 2012−11−23 19:12:13
          2012−11−23 → Fri 2012−11−23 00:00:00
            12−11−23 → Fri 2012−11−23 00:00:00
            11:12:13 → Fri 2012−11−23 11:12:13
               11:12 → Fri 2012−11−23 11:12:00
                 now → Fri 2012−11−23 18:15:22
               today → Fri 2012−11−23 00:00:00
           today UTC → Fri 2012−11−23 16:00:00
           yesterday → Fri 2012−11−22 00:00:00
            tomorrow → Fri 2012−11−24 00:00:00
tomorrow Pacific/Auckland → Thu 2012−11−23 19:00:00
            +3h30min → Fri 2012−11−23 21:45:22
                 −5s → Fri 2012−11−23 18:15:17
           11min ago → Fri 2012−11−23 18:04:22
          @1395716396 → Tue 2014−03−25 03:59:56
```

Note that timestamps displayed by remote systems with a non−matching timezone are usually not parsable locally, as the timezone component is not understood (unless it happens to be "UTC").

Timestamps may also be specified with microsecond granularity. The sub−second remainder is expected separated by a full stop from the seconds component. Example:

```
2014−03−25 03:59:56.654563
```

In some cases, systemd will display a relative timestamp (relative to the current time, or the time of

invocation of the command) instead of or in addition to an absolute timestamp as described above. A relative timestamp is formatted as follows:

2 months 5 days ago

Note that a relative timestamp is also accepted where a timestamp is expected (see above).

CALENDAR EVENTS

Calendar events may be used to refer to one or more points in time in a single expression. They form a superset of the absolute timestamps explained above:

Thu,Fri 2012−*−1,5 11:12:13

The above refers to 11:12:13 of the first or fifth day of any month of the year 2012, but only if that day is a Thursday or Friday.

The weekday specification is optional. If specified, it should consist of one or more English language weekday names, either in the abbreviated (Wed) or non−abbreviated (Wednesday) form (case does not matter), separated by commas. Specifying two weekdays separated by ".." refers to a range of continuous weekdays. "," and ".." may be combined freely.

In the date and time specifications, any component may be specified as "*" in which case any value will match. Alternatively, each component can be specified as a list of values separated by commas. Values may be suffixed with "/" and a repetition value, which indicates that the value itself and the value plus all multiples of the repetition value are matched. Two values separated by ".." may be used to indicate a range of values; ranges may also be followed with "/" and a repetition value.

A date specification may use "~" to indicate the last day(s) in a month. For example, "*−02~03" means "the third last day in February," and "Mon *−05~07/1" means "the last Monday in May."

The seconds component may contain decimal fractions both in the value and the repetition. All fractions are rounded to 6 decimal places.

Either time or date specification may be omitted, in which case the current day and 00:00:00 is implied, respectively. If the second component is not specified, ":00" is assumed.

Timezone can be specified as the literal string "UTC", or the local timezone, similar to the supported syntax of timestamps (see above), or the timezone in the IANA timezone database format (also see above).

The following special expressions may be used as shorthands for longer normalized forms:

```
    minutely → *−*−* *:*:00
      hourly → *−*−* *:00:00
       daily → *−*−* 00:00:00
     monthly → *−*−01 00:00:00
      weekly → Mon *−*−* 00:00:00
      yearly → *−01−01 00:00:00
   quarterly → *−01,04,07,10−01 00:00:00
semiannually → *−01,07−01 00:00:00
```

Examples for valid timestamps and their normalized form:

```
  Sat,Thu,Mon..Wed,Sat..Sun → Mon..Thu,Sat,Sun *−*−* 00:00:00
     Mon,Sun 12−*−* 2,1:23 → Mon,Sun 2012−*−* 01,02:23:00
               Wed *−1 → Wed *−*−01 00:00:00
         Wed..Wed,Wed *−1 → Wed *−*−01 00:00:00
             Wed, 17:48 → Wed *−*−* 17:48:00
Wed..Sat,Tue 12−10−15 1:2:3 → Tue..Sat 2012−10−15 01:02:03
            *−*−7 0:0:0 → *−*−07 00:00:00
                10−15 → *−10−15 00:00:00
```

```
monday *−12−* 17:00 → Mon *−12−* 17:00:00
Mon,Fri *−*−3,1,2 *:30:45 → Mon,Fri *−*−01,02,03 *:30:45
12,14,13,12:20,10,30 → *−*−* 12,13,14:10,20,30:00
12..14:10,20,30 → *−*−* 12..14:10,20,30:00
mon,fri *−1/2−1,3 *:30:45 → Mon,Fri *−01/2−01,03 *:30:45
03−05 08:05:40 → *−03−05 08:05:40
08:05:40 → *−*−* 08:05:40
05:40 → *−*−* 05:40:00
Sat,Sun 12−05 08:05:40 → Sat,Sun *−12−05 08:05:40
Sat,Sun 08:05:40 → Sat,Sun *−*−* 08:05:40
2003−03−05 05:40 → 2003−03−05 05:40:00
05:40:23.4200004/3.1700005 → *−*−* 05:40:23.420000/3.170001
2003−02..04−05 → 2003−02..04−05 00:00:00
2003−03−05 05:40 UTC → 2003−03−05 05:40:00 UTC
2003−03−05 → 2003−03−05 00:00:00
03−05 → *−03−05 00:00:00
hourly → *−*−* *:00:00
daily → *−*−* 00:00:00
daily UTC → *−*−* 00:00:00 UTC
monthly → *−*−01 00:00:00
weekly → Mon *−*−* 00:00:00
weekly Pacific/Auckland → Mon *−*−* 00:00:00 Pacific/Auckland
yearly → *−01−01 00:00:00
annually → *−01−01 00:00:00
*:2/3 → *−*−* *:02/3:00
```

Calendar events are used by timer units, see **systemd.timer**(5) for details.

Use the **calendar** command of **systemd-analyze**(1) to validate and normalize calendar time specifications for testing purposes. The tool also calculates when a specified calendar event would elapse next.

SEE ALSO

systemd(1), **journalctl**(1), **systemd.timer**(5), **systemd.unit**(5), **systemd.directives**(7), **systemd-analyze**(1)