

## xppq : details

---

### Output formating

---

#### Display markers

Marker	Signification
%%	% sign
%<f>d	Amount of days
%<f>D	Same as %<f>d
%<f>h	Hours in the day
%<f>H	Total amount of hours
%<f>m	Minutes in the hour
%<f>M	Total amount of minutes
%<f>s	Seconds in the minute
%<f>S	Total amount of seconds
%<f>f	Frames in the second
%<f>F	Total amount of frames
%r	Remainder as float
%R	Remainder as fraction
%t	FPS-type dependent token (see below)

<f> : [<w>][ ( . | , ) [<p>] ] (1 by default). Optional set of following flags :

- <w> : count of the minimum amount of digits for the integer part,
- . or , : the decimal separator to use,
- <p> : count of the maximum amount of digits for the fractional part.

The tokens option is used for the '%t' format marker. For drop frame, the first character is used. For non-real-time, the second character is used if set. If not defined the one for drop frame is used. For non drop frame, the third character is used if set. Alternatively the one for non-real-time is used.

#### Rounding markers

These optional markers specify how the timecode values are rounded, but don't cause anything to be displayed. You have to specify a display marker (see above) to see some output.

This rounding markers affect the timecode as a whole. For example, if you specify %<m (round down the minutes), the seconds (and frames) value will always be 0.

Marker	Signification
%<[<p>]<t>	Round down
%=[<p>]<t>	Mathematical rounding (default rounding)
%>[<p>]<t>	Round up

- <p> : rounding precision (optional, 0 to 9, 0 if missing).
- <t> : rounding target (d for days, h for hours, and so on ...)

#### FPS designations

---

- 23n (23.976 fps non-drop frame, not real-time<sup>1)</sup>)
- 29n (29.97 fps non-drop frame, not real-time<sup>2)</sup>, uncommon)
- 29d (29.97 fps drop-frame, 30/1.001<sup>3)</sup>, de-facto standard for NTSC work)
- 30d (alias of 29d)
- 59d/60d<sup>4)</sup>
- 59n/60n
- other integers,
  - especially 24, 25, 30, 50, 60, 100, 600, 1000, ... (for video),
  - especially 44100, 48000, 96000 (for audio).

## Timecode formats

---

- hh:mm:ss:ff or h:m:s:f (hours:minutes:seconds:frames)
- m:s:f
- s (seconds integer value)
- s.p(ost decimal positions)<sup>5)</sup>
- d:h (days:hours)
- frames: frame count starting at 0 or at 1 (default: 0 is the first frame, as widely used in compositing applications).

You can append a *FPS* designation (see above) by preceding it by a @. When not specified, the FPS designation used is the one specified for the output.

---

<sup>1)</sup> , <sup>2)</sup> *not realtime* means that the seconds are shorter than the seconds we know and love. Seconds, minutes and hours will drift away from what a wall clock would show. For 29n this would mean that an hour is about 3.6 seconds (3600-3600/1.001) shorter compared to real-time.

<sup>3)</sup> Timecode for frame 0 and 1 are dropped for every minute but those divisible by ten, 108 timecode values dropped per hour.

<sup>4)</sup> Same timecode dropping rule as for 29d but dropping twice as many: frame 0, 1, 2 and 3 are dropped instead of just the first two → 216 timecode values dropped per hour.

<sup>5)</sup> May make precision definition necessary.