
Dates and Times

Encoding and Calculating

This practice defines how to encode dates, times and durations; and perform calculations such as adding a duration to a date.

This applies to Common Metadata, Media Manifest and Avails (both Excel and XML).



This work is licensed under a [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/).

NOTE: No effort is being made by the Motion Picture Laboratories to in any way obligate any market participant to adhere to this specification. Whether to adopt this specification in whole or in part is left entirely to the individual discretion of individual market participants, using their own independent business judgment. Moreover, Motion Picture Laboratories disclaims any warranty or representation as to the suitability of this specification for any purpose, and any liability for any damages or other harm you may incur as a result of subscribing to this specification.

REVISION HISTORY

Version	Date	Description
1.0	November 22, 2017	Initial publication

1 OVERVIEW

Throughout Common Metadata, Media Manifest and Avails, there are fields that capture dates, times and durations. This document describes the basic uses cases, including subtleties that could trip up an implementer.

We have attempted to provide precise definitions so there is no confusion. Please let us know if there are any cases not covered.

2 DATA TYPES

Date, time and duration are expressed using the XML defined types.

XML type definitions are based on ISO 8601, described here:
https://en.wikipedia.org/wiki/ISO_8601. W3Schools has a nice explanation with examples:
https://www.w3schools.com/xml/schema_dtypes_date.asp

In Excel, dates can be expressed in Excel format or ISO format. Dates+times and durations must be specified in ISO format.

3 DATES AND TIMES

3.1 Dates and Dates+Times

We can specify dates (July 1, 2017), dates with a time of day (July 1, 2017, 2:00 PM) or just time of day (2:00 PM). Dates can be specified as just year (2017), year and day of year (July 1, 2017), or just day of year (July 1).

Generally, for dates in the past, we use year (1965) or date (July 1, 1965). For dates in the future (Avails), we use dates (July 1, 2018) or dates with time of day (July 1, 2018, 0:00:00Z).

Dates and times are encoded using the XML types `xs:date`, `xs:dateTime`, and `xs:gYear`. We have also defined a type `YearDateOrTime-type` that can be any of these.

Here are some examples.

Type	Example(s)
<code>xs:date</code>	2017-07-01Z
<code>xs:dateTime</code>	2017-07-01T14:00Z, 2017-07-01T09:00
<code>xs:gYear</code>	1965
<code>YearDateOrTime-type</code>	2017-07-01Z 2017-07-01T14:00Z 1965

3.2 UTC and Time Zones

When specifying dates and times it is necessary to have a clear understanding of the time zone. This applies not just to time, but to date.

We will address three methods for dealing with time zone:

- UTC – Universal Coordinated Time (time as an absolute)
- Specific time zone
- No time zone

Note that the first two are and absolute time reference. The latter is used to refer to local time, which requires interpretation of the time in context. For Avails, it is strongly recommended that all dates include time and that time be expressed in UTC unless specifying local time zone.

3.2.1 Specifying dates and times UTC/GMT/Zulu

UTC (in English, Universal Coordinated Time) allows you to specify time independent of time zone. Specifying a time in UTC is unambiguous as it independent of time zone and regional variations (e.g., daylight savings).

Because it is unambiguous, most of our practices recommend the use of UTC.

For the specs discussed in this document, UTC, Zulu and GMT are the same. However, they aren't actually the same and it's more correct to refer to UTC. Here's a nice article on the topic: <https://www.timeanddate.com/time/aboututc.html>

Here's an example. Note the "Z" (Zulu) that indicates this is UTC.

```
<avails:Start>2017-07-1T00:00:00Z</avails:Start>
```

3.2.2 Specifying specific time zone

Time zones can be specified as hours and minutes relative to UTC. For example,

```
<avails:Start>2017-07-1T00:00:00-08:00</avails:Start>
```

refers to 8 hours before UTC (i.e., PST). Some regions are not on 'the hour' and can offsets with minutes. For example, Newfoundland Standard Time is -03:30, and Nepal Standard Time is +05:45.

3.2.3 No time zone specified

When no time zone is specified, ISO 8601 defines this as "local time" with the note, "no provisions have been made to prevent ambiguities in expressions that result from discontinuities in the time scale of local time." To avoid this ambiguity, we define local time as 'the first time zone in the geo'. That means within the territory, the first time zone to reach that time.

Let's say a title is availed for July 1, 2017 for all off Canada which has 7 time zones, including the infamous Newfoundland Standard Time. Avail Start assumes a time of 00:00:00 when

no time is specified. The first time zone in Canada is Newfoundland. So, the title goes live 00:00:00 July 1, 2017 in Newfoundland, June 30, 2017, 23:30:00 Atlantic, June 30, 2017: 22:30:00 Eastern, and so forth.

Note that this is the equivalent of specifying a start July 1, 2017, 03:30:00 Zulu.

4 DATE AND TIME DURATION

This section addresses durations typically found in Avails. Durations of audiovisual material is addressed below.

Duration is expressed in seconds, hours, days, months and years. In this context we generally don't care about fractions of a second. First, we must precisely define what it means to express time in terms of hours, days, months and years.

4.1 What is a day, month, year?

We think about a day being 24 hours, but a day can also be interpreted as, “until the same time tomorrow”. A “month” can be “the same day next month”, but what if it's January 31? A year is 365 days, except for leap years. A year could be “same date next year”, but what about February 29.

XML allows the author to be very specific and provides direction on how to deal with exceptions. This allows you to *record what you mean*, and let the recipient perform the well-defined deterministic calculations to achieve the desired result. This allows you to differentiate between 24 hours and one day.

4.2 Calculating with dates and durations

We've established that the duration can specify exactly what the author meant (e.g., 24 hours vs. 1 day).

But, how do we add a duration to a date? It turns out this is precisely defined in <https://www.w3.org/TR/xmlschema-2/#adding-durations-to-dateTimes>

The algorithm accounts for all anomalies such leap days and seconds. The algorithm also solves date calculation problems (e.g., adding on month to March 31, resulting in April 31) with a principle they call ‘pinning’ (e.g., April 31 pins to April 30).

4.3 Avails Windows

Avails Start and End originally had only date. When a window starts on a day, it's interpreted to be midnight (morning) of that day. Windows end on midnight of the day. It is assumed that time does not roll across time zones (i.e., the window starts or ends simultaneously across the entire territory).

However, this is ambiguous. For example, when you say “midnight” in the US, does that apply to midnight on the East Coast, or midnight in each time zone (Eastern, Central, Mountain, Pacific,

Hawaii, Guam, etc.)? What about locales like Arizona where only parts of the state use Daylight Savings Time?

To avoid ambiguity, we added the term “first time in geo”. In the US, it’s Eastern. In Canada it’s Newfoundland time. The window starts at midnight (beginning of the day) across the entire territory and ends at midnight (end of the day) across the entire territory. For example, a window that starts on 1/1/18 starts at 00:00:00 EST, 12/31/17 23:00:00 CST, 22:00:00 MST and 21:00:00 PST.

Unfortunately, there are exceptions that make this problematic. For example, Newfoundland, the first territory in Canada, has a 30-minute daylight savings adjustment. So, midnight (start of day) in Newfoundland, is 11:30 PM the previous day in Atlantic, 10:30PM Eastern, etc. This is probably not the desired time. And, parts of Arizona don’t observe Daylight Savings Time. So, as specified, date alone doesn’t necessarily yield the right results.

XML is more flexible, using the ISO definition. Time without time zone is local time and it can roll across time zones.

5 AUDIOVISUAL TIMING

When we talk about audiovisual time or duration, we’re referring to the length of an entire work, a track or a clip; or a particular place in the middle of one of those things.

5.1 Close is often good enough

In many cases, close is good enough. For example, when presenting the length of an episode to present to a user, the precise number of seconds or frames isn’t necessary (or even known). It’s good to get it as close as possible, but sometimes metadata must refer to multiple regional edits, all off by a little bit. It’s all good—precision is not important.

5.2 Frame accurate timing

In other cases, frame accurate timing is necessary. For example, entry and exit points, and bonus time synchronization should be frame accurate. This is well understood art, which means there are solutions.

Times in audiovisual material expressed as ‘timecodes’. Timecodes are not easy—entire books have been written on timecodes—but we have a simplified model that addresses most (all?) use cases to frame accuracy.

Some timecode systems encode frames (e.g., 234.12 means the 12th frame of second 234). If this content is transcoded with 3:2 pulldown (30fps), that timecode is longer valid. As metadata is generally neutral to encode, we must avoid issues of frame rate. There’s a whole section on this in the Media Manifest spec (Section 2.3.1) which describes how this is done.