

# **MovieML (Beta)**

Version 00.45b

**22<sup>nd</sup> November, 2020**

## CONTENTS

1. Summary.....	3
2. Background.....	3
3. Technology .....	3
4. Development History .....	3
5. Key Concepts .....	4
6. Using MovieML.....	5
7. Conclusion .....	8
References .....	9

# MovieML (Beta)

Last updated: 22<sup>nd</sup> November 2020

## 1. Summary

MovieML (Beta) is primarily a web app for setting markers in videos and audio tracks for use as a media mark-up tool for creators, hobbyists, theorists, and presenters. It can be used for setting simple bookmarks in favourite films or audio tracks, conducting detailed sequence-by-sequence annotations of media files, and extracting timecourse segments to video and audio clips.

Markers can be set either individually as single timestamps, or in pairs to bracket runtime segments. Marker-sets can be created, colour-coded and annotated per media file, as well as being organised into generic sequences and exported as XML-formatted schemas. Runtime segments can also be exported as video clips.

## 2. Background

Statistical and experimental software for analysing film timelines range from shot-length analysis (eg. [Cinemetrics](#)) and eye-tracking applications ([OpenGazer](#), [PyGaze](#)) to emerging machine-learning environments ([StoryFit](#), [Cinelytic](#) etc). MovieML fills a niche by allowing digital timecourse-media assets to be ‘marked-up’ in [XML](#) in much the same way that XML-based mark-up notations can be applied to text.

## 3. Development History

Pre-alpha requirements were shaken out over the course of a research project where a need was identified for a tool which could be used to sequence and annotate plot segments in video files and export them in readily convertible formats. An initial feature-set was frozen in the summer of 2020, and a feature-complete pre-beta codebase drafted by mid-November of the same year. At the time of writing, beta trials are underway with an informal group of testers.

## 4. Technology

MovieML is a browser-based web app accessed from <http://movieml.org>. In its native web form, it can be used to set and save video marker information directly from media files on a user’s hard disk

without needing to upload the files to a server. The executive source code is written in Javascript ([ECMAScript2019](#)), and presentation mark-up and styling are rendered in [HTML5](#) and [CSS3](#) respectively. Chrome and Edge browsers are recommended as they both support a wider range of native media filetypes than Firefox. A minimum screen resolution of 1280 by 720 is also recommended.

For extracting and saving video clips, the user must install a patch, available from <http://MovieML.org/downloads/>. This functionality is implemented for Windows only in the beta version, but will ultimately include Apple and Linux. The current download package includes an [FFmpeg](#) distribution, an intermediate link-processing executable called ‘caller.exe’, and a registry patch. FFmpeg calls from the online MovieML code are executed locally as a custom URI protocol via caller.exe. The registry patch registers caller.exe as a custom URI handler. Installation instructions for the video clip patch are available from <http://MovieML.org/help/>.

## **5. Key Concepts**

### **Marker**

Markers are MovieML’s basic building blocks. New markers are created with a single timestamp field. Each marker contains a note, and can be colour-coded. Clicking on the marker button moves the media player’s seekbar to the marker’s timestamp. Single markers can be combined with split markers for export to an MMLMovie file, but cannot be included in an MMLSchema (covered later).

### **Split Marker**

A single marker can be split into a pair using the ‘Split’ function in the ‘Edit’ screen. Split markers can be used to bracket a timecourse segment, and be exported as part of an MMLSchema marker set.

### **Saving Files**

Session data, comprising new and/or imported marker data, is held in memory and not saved until the ‘Save to File’ button is clicked. MovieML uses the browser’s download functionality to mimic the ‘Save As’ functionality of a desktop app, and files are therefore saved to the browser’s default download destination. Browser settings can be modified to specify different save destinations if required.

## MMLMovie file

All session markers, single and split, and including timestamps, colour coding, and notes, are exported as an XML-formatted MMLMovie file when the ‘Save to File’ button is clicked. This file is created as a mediafile-specific document, with a header containing automatically-generated mediafile metadata and a body containing the marker information. A warning will be generated if this MMLMovie file is subsequently loaded into a session using a different mediafile, but its markers can nonetheless be merged with that session if required. If details are merged in this way, the header metadata from the incoming MMLMovie file are discarded in favour of those generated by the currently active mediafile.

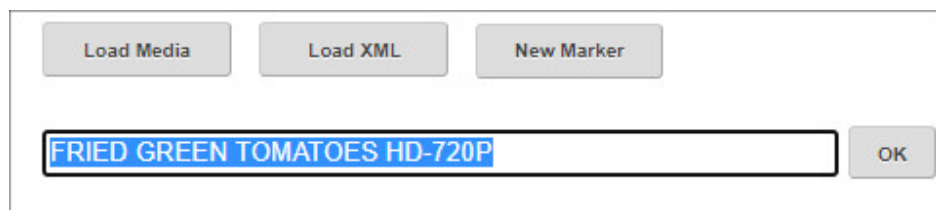
## MMLSchema file

The ‘Export Schema’ button saves an XML-formatted MMLSchema file to disk. This file contains a header with information about the schema, and a list of the named split markers ordered by ascending start times. Singleton markers are discarded, as are mediafile metadata, timestamps, notes and button liveries, as the purpose of the schema is to present a generic marker set for application across different media files. When loaded into a session using the ‘Load XML’ button, schema markers are merged with the existing session data as a new marker set with empty notes, timestamps, and button livery fields which can be edited and exported to a session-specific MMLMovie file.

## 6. Using MovieML

The examples used in this section refer to video files, but can be applied to any timecourse mediafile recognised by the browser’s HTML5 media player.

### Opening a Media File



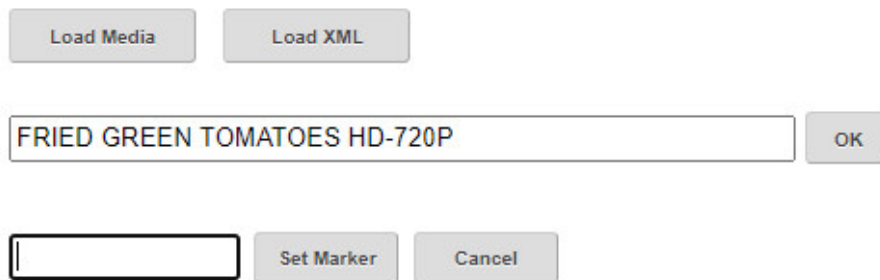
The image shows a web interface for opening a media file. At the top, there are three buttons: 'Load Media', 'Load XML', and 'New Marker'. Below these buttons is a text input field with the text 'FRIED GREEN TOMATOES HD-720P' and an 'OK' button to its right.

When a new video file has been loaded using the ‘Load Media’ button, a video title is automatically generated from the filename. Change the name and click ‘OK’, or skip this step if an MMLMovie

file is about to be loaded which already contains a name for this video.

## Creating Markers

Use the ‘Load XML’ button to load either an MMLMovie or an MMLSchema file, or click the ‘New Marker’ button to create a new marker.



The image shows a UI for creating a marker. At the top, there are two buttons: 'Load Media' and 'Load XML'. Below them is a text input field containing the text 'FRIED GREEN TOMATOES HD-720P', followed by an 'OK' button. At the bottom, there is a text input field (which is empty), a 'Set Marker' button, and a 'Cancel' button.

Marker names must be unique to the session, unspaced, alphanumeric, and contain no more than 22 characters. The underscore ‘\_’ character can be used instead of a space. Type in a marker name and click ‘Set Marker’. A new singleton marker is created in the default blue livery. The marker timestamp will be set automatically to the media player’s current seekbar time.

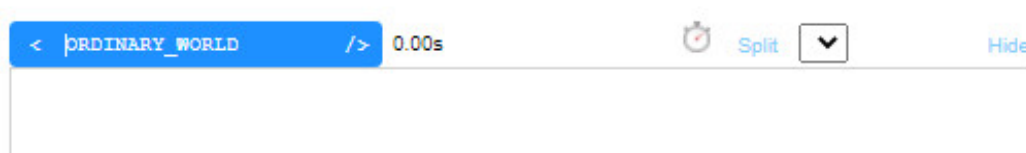


The image shows a UI element representing a created marker. It consists of a blue button with the text '< ORDINARY\_WORLD />' and '0.00s' to its right. To the right of the button are three links: 'Expand', 'Edit', and 'Clear'. Below this, there are two buttons: 'Save to File' and 'Export Schema'.

New markers can also be created using the ‘</>’ icon on the media player controls.

## Editing a Marker

Clicking on the blue ‘Edit’ link to the right of the marker button opens the marker’s edit controls.

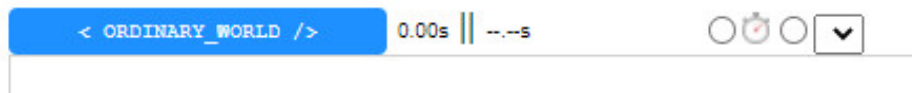


The image shows the UI for editing a marker. It features a blue button with the text '< ORDINARY\_WORLD />' and '0.00s'. To the right of the button are icons for a clock, a 'Split' link, a dropdown menu, and a 'Hide' link. Below this is a large, empty rectangular text area for editing the marker name.

Edit the marker name in the button text box.

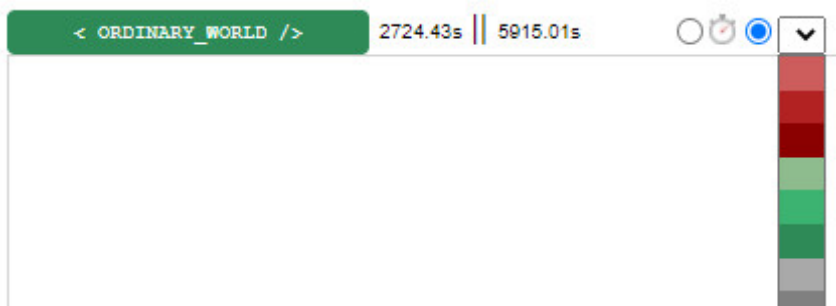
To change the timestamp, move the media player seek bar to the desired time and click on the 🕒 icon.

Split the marker and add an end timestamp by clicking on the blue ‘Split’ link (shown in the previous illustration) which produces:

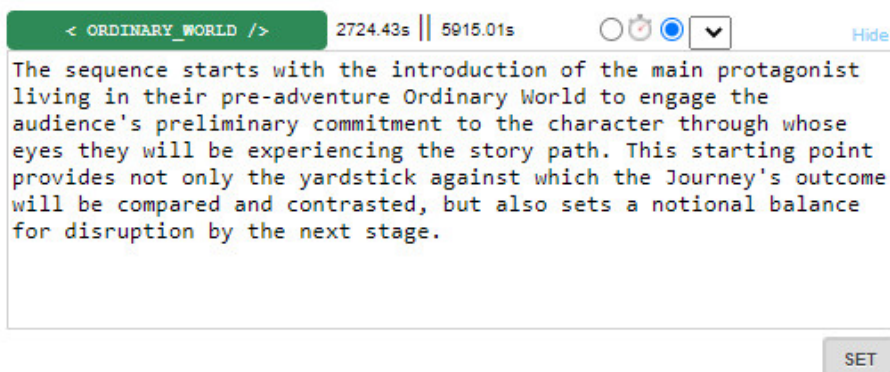


The end timestamp can be set by moving the media player seekbar to the desired end time, selecting the radio button to the right of the stopwatch icon, and then clicking the stopwatch icon. The start time can be reset in the same way using the left-hand radio button.

The button livery can be changed using the dropdown to the right of the right-hand radio button:



Annotations can be added in the textarea (up to 1000 characters):

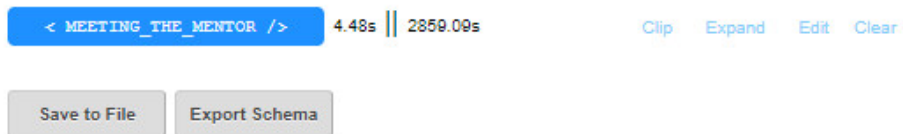


To add the changes to the session data, click the ‘SET’ button. (This adds your changes to the session data only, and does not save them to disk.)

Marker notes can be viewed by clicking on the blue ‘Expand’ link.

Additional marker sets can be imported using the ‘Load XML’ button.

Split markers can be exported as video clips using the ‘Clip’ link if the video clip patch has been installed (available from <http://MovieML.org/downloads>). The source video must be copied into the <C:\MovieML\media> directory before the MovieML session is started and loaded using the ‘Load Media’ button. Clips are extracted to <C:\MovieML\media\clips>.



## CAUTION

Session data is held in memory, and not saved until the ‘Save to File’ button is clicked.

## 7. Conclusion

MovieML is still undergoing early-stage testing, is likely to contain bugs and unexpected behaviours, and should not be relied upon for real-world applications.

Bug reports are gratefully received, and new tickets should be filed at:

<https://movieml.org/ts/>

## References

Cinelytic, <https://www.cinelytic.com/>

Cinemetrics, <http://cinemetrics.lv/>

CSS, <https://www.w3.org/Style/CSS/>

ECMAScript 2019, <https://www.ecma-international.org/ecma-262/10.0/index.html>

EEGLAB - Swartz Center for Computational Neuroscience, <https://sccn.ucsd.edu/eeglab/>

Extensible Markup Language (XML), <https://www.w3.org/XML/>

FFmpeg, <https://ffmpeg.org/>

HTML 5.2, <https://www.w3.org/TR/html52/>

MatLab, <https://www.mathworks.com/products/matlab.html>

Opengazer: a webcam-based eye tracker, <https://sourceforge.net/projects/opengazer/>

PyGaze: Open source eye-tracking software, <http://www.pygaze.org/>

StoryFit: AI for the Entertainment Industry, <https://www.storyfit.com/>