Motion Browser

Nigel D. Pegram

June 15, 2019

Contents

1 Readme

Browser-based user interface for the Motion application. Developed by Carlos Ladeira (caladeira@gmail.com), Updated by Nigel D. Pegram (ndpegram@gmail.com). This software is distributed under the GNU public license.

Tested with Motion 4.0

This web page reads the mysql database filled by Motion and output events by day It creates small thumbnail were you can click to watch the movie file for the same event. You can also delete the events you select.

It's a good interface in case you are running Motion on a computer without monitor/keyboad/mouse, only network!

It is supposed to work with the following motion.conf options set as shown (for better results):

```
pre_capture 8 (works for me with a framerate of 6)
post_capture 8 (ie)
output_all off
output_normal best (or) first
output_motion off
text_event %Y%m%d%H%M%S
ffmpeg_cap_new on
ffmpeg_video_codec msmpeg4
sql_log_image on
sql_log_snapshot off
sql_log_mpeg on
sql_log_timelapse off
```

```
sql_query INSERT ... (I use the default)
mysql_db motion (my database name)
mysql_host localhost
mysql_user ... (the user name i created in MySQL)
mysql_password ... (the password associated with user)
```

2 Installation

2.1 File system

Install the files into the appropriate location in your web software's file tree.

The directory where you store your motion files must be writeable by the user under which the web software is running. In Ubuntu, for example, this is the `www-data` user. For example, if the directory to which motion is saving the video and image files is `/var/lib/motion`, then you should issue the following (assuming the web user is `www-data`). You will likely need to issue this commands as a superuser.

```
sudo chgrp -R www-data /var/lib/motion
sudo chmod -R g+rw /var/lib/motion
```

2.2 mySQL

Use the following to create your mySQL table.

```
CREATE TABLE ‘security‘ (
  ‘camera‘ int(11) DEFAULT NULL,
  ‘filename‘ varchar(80) NOT NULL DEFAULT ’’,
  ‘frame‘ int(11) DEFAULT NULL,
  ‘file_type‘ int(11) DEFAULT NULL,
  ‘time_stamp‘ timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMES
  ‘text_event‘ varchar(40) NOT NULL DEFAULT ’0000-00-00 00:00:00’,
  ‘event_time_stamp‘ timestamp NOT NULL DEFAULT ’2000-01-01 00:00:00’,
  ‘file_size‘ varchar(36) NOT NULL DEFAULT ’0’,
  KEY ‘time_stamp‘ (‘time_stamp‘),
  KEY ‘event_time_stamp‘ (‘event_time_stamp‘)
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```

2.3 Apache

You need to set the appropriate values in the security.conf apache settings file.

ServerName should be set to the name under which you will serve the site.

ProxyPass and ProxyPassReverse URLs should be set to the address where motion is running. This address should use dotted quad notation and include the port.

For example: `http://192.168.2.42:9080/`, where 192.168.2.42 is the server where the motion service is running and 9080 is the port where motion is running (see `webcontrol_port` in the `motion.conf` file.).

DocumentRoot should be set to the fully-qualified path of where you intalled the MotionBrowser files

Other Log names and locations can be set as required.

3 History

3.1 Version 1.2

20190602 Largely internal reorganisation

- Converted from GET to POST, to allow deletion of selection of large number of videos on one day. This includes removal of overarching form in HTML code and using javascript to post and AJAX instead.
- Converted from PHP include files to gettext for internationalisation.
- Use AJAX for integration of internationalisation of javascript messages (alert/confirm) with PHP gettext calls. Thus only one set of internationalisation files are required.
- Miscellaneous bug fixes and enhancements.

3.2 Version 1.1

20190422 Adapted by Nigel Pegram.

- Updated to PHP 7 and mySQL 14
- Adjusted to stream video files rather than download
- Miscellaneous bug fixes.

3.3 Version 1.0

20060000 Original Carlos Ladeira version.

4 To Do/Planned features

- Update the code to object-oriented format
- Add link to video preview "box" to allow downloading of file, rather than streaming. (Is this necessary since we can do this from the stream?)
- Add button to delete data and files based on user-selected date.
- Building on the above, add settings and code to respond to disk free space falling below a preset level. This should trigger a warning/dialog, which the user can then set a date up to which all files and database rows are deleted. There are a number of possible ways to implement this:
  1. Insert a date into a file which a cron script reads and acts on during low load periods.
  2. Delete from within the web browser (problematic as it will tie up the browser when, presumably, the user will want to interact with it).
  3. Use a hybrid approach. Delete the database rows immediately, but write out the files to delete for later crontab processing. The advantage of this approach is that the items will no longer display in the browser and the interface should remain responsive.

     Question: where to save the list of text files? Possible locations are /etc/motion or the files directory. The latter seems preferable. It seems wise to set this as a variable in the config file which defaults to the files directory.