Installing kmotion

Howto install kmotion

Installing kmotion

Server requirements ...

kmotion has been developed and tested on ubuntu 7.10. Although kmotion has modest requirement your mileage may vary on different distros. Here is a summary of the key server requirements.

Motion	v3.2.3
Apache2	v2.2.4
Libapache2-mod-php5	v5.2.3
Disk space	min 5 GB

The amount of disk space that kmotion uses can be adjusted. See the FAQ

Using the automated install script ...

Download the tarball and unzip it. Decide where you want the kmotion directory to live and move it there. kmotion can be installed almost anywhere but $HOME is as good as any.

```
cd kmotion
sudo ./install.py
.....
<press enter to start the install>
.....
```

And with any luck kmotion has installed. If everything is showing green [OK]'s goto the 'configuring kmotion' section.

Manually installing ...

So you want do do it the hard way huh ?

Download the tarball and unzip it. Decide where you want the kmotion directory to live and move it there. kmotion can be installed almost anywhere but $HOME is as good as any.

```
cd kmotion/daemons
./install_int.py
lpr daemon.rc
```

'install_int.py' sets the correct paths for the internal directories, generates a 'kmotion_vhost' file with the correct paths and generates 'kmotion' and 'kmotion_restart' with the correct paths. If you can print the daemons.rc it will be usefull for later reference.

```
sudo apt-get install libapache2-mod-php5 apache2 motion ntp screen
```

Although ntp and screen are not strictly neccesary if you are using a headless server they are a must.

```
sudo ln -s <apache2_config_dir from daemon.rc>/kmotion_vhost  /etc/apache2/sites-enabled
sudo /etc/init.d/apache2 restart
```

Link 'kmotion_vhost' to apache2 and restart apache2.

```
sudo mv kmotion /usr/bin/
sudo mv kmotion_restart /usr/bin
```

'kmotion' and 'kmotion_restart' are 'bin' files that have the correct paths to access the kmotion daemons. They were generated by 'install_int.py'.

```
sudo vi /etc/rc.local
```

Add the line 'sudo -u <your loggon name> kmotion &' to the line above 'exit 0' in '/etc/rc.local'. This starts kmotion on a reboot.

```
sudo vi /etc/hosts
```

Add the line '127.0.0.1   kmotion' as the first line of the file. This allows 'http://kmotion' to be routed through to apache2's 'kmotion_vhost'.

And with any luck kmotion has been installed. If everything went smoothly goto the 'configuring kmotion' section.

Configuring kmotion

How to setup motion.conf for kmotion

Configuring kmotion

kmotion uses the same configuration file as motion ie 'motion.conf'. 95% of all configuration for kmotion is the same as motion with a few exceptions.

In order for kmotion to display live jpegs from motion and offer event playback motion has to be configured in a certain way.

(1) Firstly motion.conf must be configured in thread mode even if you only have one camera.

(2) Secondly kmotion has to override five motion.conf options. Namely: jpeg_filename, snapshot_filename, on_event_start, on_event_end, in_picture_save and target_dir.

(3) Last an extra option. To get kmotion to display the cameras description in the top LHS use the kmotion option '#ktext <text>' inside 'motion.conf' or its threads. So if camera 3 is of a garage, in thread 3 you would add the line:

#ktext messy garage

So now you know the restrictions, go configure motion.conf as you normally would eg setting 'snapshot_interval', I use 500 secs, and 'framerate', I use 4. kmotion looks for 'motion.conf' in the same places motion does./motion.conf, /etc/motion/motion.conf, ~/.motion/motion.conf
and /usr/local/etc/motion.conf

When done do not start motion just type 'kmotion'. If you change 'motion.conf' type 'kmotion_restart' to get kmotion to reconfigure.

Finaly point your browser (hopefully firefox) to 'http://kmotion', the default username is 'kmotion' the default password is 'kmotion' and have fun !

FAQ

Frequently asked Questions.

FAQ

'Its not working   are there any diagnostic tools ?'

cd to kmotion/daemons. Execute 'daemon_manual.py'. This gives you manual control and status information of the daemons.

Execute 'tail -f /var/log/messages' to get real time reporting from both kmotion and motion. To increase the verbosity of kmotion change the reporting level of the daemons as below.

'How can I change the reporting level of the daemons ?'

cd kmotion/daemons and edit 'daemon.rc'. Look for the subsection '[debug]' and the option 'log_level = WARN'. Change this value to EMERG, ALERT, CRIT, ERR, WARNING, NOTICE, INFO or DEBUG. Then manually kill all kmotion processes and restart with 'kmotion' (or just reboot). Live logging can be seen by executing 'tail -f /var/log/messages'

'How do I change the default user name and password ?'

cd to the kmotion/apace2_config. The passwords are held in the 'users_digest' file. To change or add passwords see 'man htpasswd'. When you have changed the password you need to restart apache2

'How can I change where the image database is situated ?'

cd to kmotion/daemons edit 'daemon.rc'. Look for the subsection '[dirs]' and the option 'images_dir = .../kmotion/images'. Change the path to point to your desired location. Then execute 'kmotion_restart'

'How can I change where directory X is situated ?'

In exactly the same way as you change where the image database is situated however if you move the daemons directory you will have to re 'sudo install.py'.

'How can I change the maximum size of the images database ?'

cd kmotion/daemons edit 'daemon.rc'. Look for the subsection '[storage]' and the option 'size_gb = 5'. Change this value to your new maximum size in GB. When the image database grows to 90% of this value the oldest dated jpegs are deleted. Finally execute 'kmotion_restart'

'What else can I change in daemon.rc ?'

Nothing !

'How can I change the contents of the kmotion_vhost file ?'

The 'kmotion_vhost' file is generated by kmotion from the 'kmotion_vhost_template' file. cd kmotion/apache2_config and edit 'kmotion_vhost_template' respecting %xxxx% strings. These strings are expanded to the correct paths when 'kmotion_vhost' is generated by 'install_int.py'. Execute 'kmotion_restart' and finally restart apache by executing 'sudo /etc/init.d/apache2 restart'.

'What is in the misc_config directory ?'

The 'misc_config' directory contains configs and files generated and used by kmotion. Do not change anything in this directory.

'Why are you not using plugin X to show video ?'

In order for kmotion to be viewable on any browser anywhere in the world it must rely on nothing more exotic than Javascript. Locked down cyber cafes will not let you execute a .jar file !

This being the case kmotion requires motion to save its recordings as a sequence of jpegs rather than video. This has two benefits, firstly any browser anywhere can view the images and secondly kmotion archive viewer can playback events at variable speed and in both directions.

Developers Information

An overview of kmotions image database structure, python back end & PHP/Javascript front end.

The images database

kmotion uses a custom database structure consisting of nested directories to store motion generated jpegs.

The images top level directory .../images/ contains

Multiple date directory's in the form 'YYYYMMDD'. kmotion catalogues all images at this level by their date of capture.

An 'events' directory. This directory holds flags that are created and deleted as motion is detected and cleared. The flags consist of number filenames with zero length. For example if the 'events' directory contains '2' and '5' motion has detected motion in threads 2 and 5. The flags are created and deleted by motions 'on_event_start' and 'on_event_end' directives.

A soft link called 'lastsnap.jpg'. This is a link generated by motion to point to the latest jpeg. It is not used by kmotion.

The images first nested directory .../images/YYYYMMDD/ contains

Multiple thread directories in the form 'NN'. kmotion catalogues all images at this level by their thread or feed number as a two digit number.

A file called 'size' containing an integer number. This number is the latest estimate of the number of bytes held in the current directory. The current days 'size' is updated every 15 mins by 'kmotion_hkd1.py'. The value of each 'size' file in each of the available date directories is added together to decide if culling of the oldest dated directories is necessary. This is also performed by 'kmotion_hkd1.py'

The images second nested directory .../images/YYYYMMDD/NN/ contains

A file called 'last_jpeg' containing a path and filename. This points to the last jpeg for this particular thread or feed. It points to a jpeg in either the 'tmp' or 'video' directories. This file is updated by motions 'on_picture_save' directive. It is used by the PHP/Javascript front end to refresh the display with the latest jpeg.

A file called 'journal_snap' containing a coded string with the format #<start second>$<interval seconds> ie #193735$300, possibly repeated several times. This equation defines the motion snapshot jpeg filenames. ie #193735$300 would mean that the following jpegs exist 193735.jpg, 194035.jpg, 194335.jpg etc (An exception being if a video directory has already been created, see below). 'journal_snap' is updated by 'kmotion_hkd2'. Multiple instances of coded strings in 'journal_snap' occur when kmotions configs have been re-loaded.

The images third nested directory .../images/YYYYMMDD/NN/tmp contains

A 'tmp' directory. This directory contains between 4 and 6 ... 8 jpegs and acts as a buffer. motion snapshots are generated every second and stored 'tmp' by the 'snapshot_filename' operator. The 'last_jpeg' file may well point to a file in 'tmp'. 'kmotion_hkd2' 'sanitises' the time stamps on these jpegs and, providing there are no video directories already created and the snapshot interval is correct, moves the jpegs to the video directory. The 'sanitisation' ensures that the 'journal_snap' equations predicted jpegs always exist. In addition the buffering allows motions 'jpeg_filename' to lay down video directories first before 'kmotion_hkd2' attempts to move the jpeg and it ensures that any jpegs pointed to by 'last_jpeg' are around long enough to be used by the PHP/Javascript front end.

The images third nested directory .../images/YYYYMMDD/NN/video contains

Multiple time directory's in the form 'HHMMSS' and multiple jpegs in the form 'HHMMSS.jpg'. The jpegs are motion snapshots every interval seconds as defined in the 'journal_snap' coded string. jpegs whose filenames clash with a directory in the form 'HHMMSS' are skipped.. The 'last_jpeg' file may well point to a file in 'video'.

The images fourth nested directory .../images/YYYYMMDD/NN/HHMMSS contains

Multiple jpegs in the form NN.jpg. These are created by motions 'jpeg_filename' directive and represent high frame rate 'video' when motion has been detected. The 'last_jpeg' file may well point to a file in 'HHMMSS'.

The Python back end

kmotion uses two python daemons as a back end to perform housekeeping duties, they are called 'kmotion_hkd1.py' and 'kmotion_hkd2.py'. The hkd part standing for House Keeping Daemon !

'kmotion_hkd1'

'kmotion_hkd1.py' is the master daemon running on a 15 minutes cycle. It calculates the size of today's jpegs in the images database and updates the 'size' file for today. It then adds the values of all 'size' files for all available days and culls the oldest images if the database has grown too large.

In addition 'kmotion_hkd1.py' checks that 'kmotion_hkd2.py' and 'motion' are both running, restarting them if for any reason they have ceased.

On a SIGHUP 'kmotion_hkd1.py' reloads its config from 'rc.daemon'

'kmotion_hkd2'

'kmotion_hkd2.py' checks and creates critical directories in the images database on a date change. These files would normally be created by 'motion' but it is possible that 'kmotion_hkd2.py' runs before 'motion' has a chance to create them..

'kmotion_hkd2.py' creates the 'journal_snap' files as detailed in the images database section. It deletes, moves or copies jpegs from the 'tmp' directory's to the 'video' directory's depending on the requested snapshot interval providing a 'sanitised' database consistent with the 'journal_snap' file.

On a SIGHUP 'kmotion_hkd2.py' reloads its config from 'daemon.rc' and updates 'journal_snap'
On a SIGKILL 'kmotion_hkd2.py' updates 'journal_snap' with a dummy string representing no further images are stored here ie #HHMMSS$86400. See the images database section for details.

Generating configs ...

kmotion can be installed anywhere on your system. One of the side effects of this is that certain config files have to be generated by kmotion on startup. The one constant config file is 'daemon.rc' which is located in the daemons directory and accessed by the daemons with a relative path './'

'gen_start_sh' generates the 'kmotion_start.sh' bash script that can be called from anywhere, it cd's to the daemon directory and calls 'daemon_start.py'

'gen_vhost' reads 'daemon.rc' and creates a vhost file 'kmotion_vhost' from 'kmotion_vhost_template' with paths expanded as defined in 'daemon.rc'

'gen_rc_motion' reads in both 'daemon.rc' and 'motions' 'motion.conf'. It generates 'www.rc' for the PHP/Javascript front end and a modified version of 'motion.conf' for 'motion' to execute complete with paths expanded as defined in 'daemon.rc'

Tools ...

There is one useful command line tool that I use to debug the back end.

'daemon_manual.py' gives you control of the daemons and reports on the daemons status, you can start, stop and reload their configs. In addition motion status lines show if there is a 'motion.conf' error.

/var/log/mess ??

The PHP/Javascript front end