

# ZONEMINDER

v1.21.0

Open Source Linux Video Camera Security

<http://www.zoneminder.com>

# CONTENTS

<b>1. Introduction.....</b>	<b>4</b>
<b>2. Requirements.....</b>	<b>5</b>
<b>3. Building .....</b>	<b>6</b>
<b>4. Installation.....</b>	<b>8</b>
<b>5. Installing from RPM.....</b>	<b>11</b>
<b>6. Tutorial .....</b>	<b>12</b>
6.1. Defining Monitors.....	12
6.2. Defining Zones.....	18
6.3. Viewing Monitors .....	20
6.4. Controlling Monitors.....	21
6.5. Filtering Events .....	21
6.6. Viewing Events .....	22
6.7. Options and Users .....	23
<b>7. Camera Control.....</b>	<b>25</b>
7.1. Control Capabilities.....	25
7.2. Control Scripts .....	27
<b>8. Mobile Devices.....</b>	<b>29</b>
<b>9. Troubleshooting .....</b>	<b>30</b>
<b>10. Change Log .....</b>	<b>33</b>
10.1. Release 1.21.0.....	33
10.2. Release 1.20.1.....	33
10.3. Release 1.20.0.....	34
10.4. Release 1.19.5.....	35
10.5. Release 1.19.4.....	37
10.6. Release 1.19.3.....	37
10.7. Release 1.19.2.....	38
10.8. Release 1.19.1.....	39
10.9. Release 1.19.0.....	40
10.10. Release 1.18.1.....	41
10.11. Release 1.18.0.....	41
10.12. Release 1.17.2.....	43
10.13. Release 1.17.1.....	44
10.14. Release 1.17.0.....	44

10.15. Release 0.9.16.....	45
10.16. Release 0.9.15.....	47
10.17. Release 0.9.14.....	47
10.18. Release 0.9.13.....	48
10.19. Release 0.9.12.....	48
10.20. Release 0.9.11.....	50
10.21. Release 0.9.10.....	51
10.22. Release 0.9.9.....	52
10.23. Release 0.9.8.....	53
10.24. Release 0.9.7.....	54
10.25. Release 0.0.1.....	54
<b>11. To Do.....</b>	<b>55</b>
<b>12. Bugs.....</b>	<b>56</b>
<b>13. Non-Bugs.....</b>	<b>57</b>
<b>14. License.....</b>	<b>58</b>

## 1. Introduction

Welcome to ZoneMinder, the all-in-one Linux GPL'd security camera solution.

A while back my garage was burgled and all my power tools were stolen! I realised shortly after that if I'd just had a camera overlooking the door then at least I'd have know exactly when and who did the dirty deed. And so ZoneMinder was born. It's still a baby but hopefully it can grow up to be something that can be genuinely useful and maybe one day either prevent similar incidents or perhaps bring some perpetrators to justice.

ZoneMinder is designed around a series of independent components that only function when necessary limiting any wasted resource and maximising the efficiency of your machine. A fairly ancient Pentium II PC should be able to track one camera per device at up to 25 frames per second with this dropping by half approximately for each additional camera on the same device, additional cameras on other devices do not interact so can maintain this frame rate. Even monitoring several cameras still will not overload the CPU as frame processing is designed to synchronise with capture and not stall it.

As well as being fast ZoneMinder is designed to be friendly and even more than that, actually useful. As well as the fast video interface core it also comes with a user friendly and comprehensive PHP based web interface allowing you to control and monitor your cameras from home or even at work or on the road. It supports variable web capabilities based on available bandwidth. The web interface also allows you to view events that your cameras have captured and archive them or review them time and again, or delete the ones you no longer wish to keep. The web pages directly interact with the core daemons ensuring full co-operation at all times. ZoneMinder can even be installed as a system service ensuring it is right there if your computer has to reboot for any reason.

The core of ZoneMinder is the capture and analysis of images and there is a highly configurable set of parameters that allow you to ensure that you can eliminate false positives whilst ensuring that anything you don't want to miss will be captured and saved. ZoneMinder allows you to define a set of 'zones' for each camera of varying sensitivity and functionality. This allows you to eliminate regions that you don't wish to track or define areas that will alarm if various thresholds are exceeded in conjunction with other zones.

ZoneMinder is fresh off the keyboard and so comes with no warranty whatsoever, please try it, send your feedback and if you get anything useful out of it let me know.

ZoneMinder is free but if you do get ZoneMinder up and running and find it useful then please feel free to visit <http://www.zoneminder.com/donate.html> where any donations will be appreciated and will help to fund future improvements to ZoneMinder. This would be especially appreciated if you use ZoneMinder as part of your business or to protect your property.

## 2. Requirements

ZoneMinder needs a couple of things to work.

Firstly, it uses MySQL so you'll need that. In order to compile you need to make sure you have a development installation and not just a runtime; this is because it needs to use the MySQL header files. If you are running an RPM based distribution then it's probably worth installing all the pure mysql rpm files to be sure you have the right ones.

Next it does things with JPEGs so you'll need at least libjpeg.a which I think come as standard nowadays with most distributions. It also uses the netpbm utilities in a very limited way to generate thumbnails under certain circumstances though this can be modified.

ZoneMinder can generate MPEG videos if necessary, for this you'll need either ffmpeg (recommended) or the Berkeley MPEG encoder (mpeg\_encode). If you don't have either, don't worry, as the options will be hidden and you'll not really miss too much. Some of the authentication uses openssl MD5 functions, if you get a grumble about these during configuration all it will mean is that authentication won't be used for streaming. The web interface uses PHP and so you need that in your apache or other web server as well, make sure MySQL support is available either statically or as a module. There are also various perl modules that you may need that vary depending on which options you choose on installation, for more details see later in this document.

Finally, there is quite a bit of image streaming in the package. So if you don't have Netscape or another browser that supports image streaming natively I recommend you get the excellent Cambozola java applet from <http://www.charliemouse.com/code/cambozola/> which will let you view the image stream in Internet Explorer and others. Otherwise you're limited to just refreshing still images.

Hardware-wise, ZoneMinder has been used with various video and USB cameras with the V4L interface. I don't have a lot of cameras myself so I've not had change to test it with a huge range personally however there is a list of devices that are definitely known to work on the web site. Please let me know if your camera works and is not listed. You do need to have Video4Linux installed. I've not got too many machines so I've only really used it on various RedHat/Fedora distributions, which seem to have everything there by default I think. SlackWare does need a bit more tinkering than other distributions; there is a document on the web site describing what users have had to do to get it working. Please give me feedback on other distributions not listed on the site.

### 3. **Building**

The first thing you need to do is run the included configure script to define some initial configuration, just type

```
./configure --with-mysql=<your MySQL root> --with-webdir=<your web directory> --with-cgidir=<your cgi directory>
```

where *--with-mysql* identifies the root directory where you have installed MySQL (usually /usr), *--with-webdir* is the directory to which you want to install the PHP files, and *--with-cgidir* is the directory to which you want to install CGI files. These directories could be /var/www/html/zm and /var/www/cgi-bin for example. If you want to use real MPEG based streaming you will need to have built and installed the ffmpeg tools. You can then also use *--with-ffmpeg=<path to ffmpeg root>* to help configure find it if it's not installed in a default location. Note, you have to make sure you have installed the ffmpeg headers and libraries rather than just the binaries, or a development package with them in. Additionally if you have built ffmpeg with the mp3lame feature turned on you may additionally need to tell configure where to find that the mp3lame library, to prevent unresolved dependencies. To do this add the *--with-lame=<path to lame directory>* option as well. There are also two further arguments you can add if your web user and group are not both 'apache'. These are *--with-webuser* and *--with-webgroup*. Type

```
./configure --help
```

for details on these options.

That's the build configuration sorted out. The next thing you have to do is do a little more runtime specific configuration. ZoneMinder configuration is scattered around various files in the distribution so to make things easier for you there is a ZoneMinder configuration utility included. Type

```
perl ./zmconfig.pl
```

to get it started. It is an interactive utility and will prompt you by asking you various questions. For most questions typing '?' will give you additional help if you need it. Once you've answered all the questions it will write out a configuration file called 'zmconfig.txt' and then process various files to substitute the information in them. If you run it again it will remember your answers from. If you just want to rerun the substitutions you can run zmconfig.pl in non-interactive mode by typing

```
perl ./zmconfig.pl -noi
```

which will just read your file and do the substitutions with no questions asked. There are two classes of options, 'core' options which much be specified with zmconfig which detail things such as database passwords which are compiled into ZoneMinder and other options with are stored in the database and which can be modified dynamically via the 'options' section of the web interface. Only the first set need to be completed with zmconfig at this stage. If you want to change just a few options and can't access the options dialog via the web you can append them as parameters to zmconfig and it will just ask you about those. So for example,

```
perl ./zmconfig.pl ZM_STRICT_VIDEO_CONFIG
```

however it is fairly dumb and will not tell you if you make a typo and misspell an option.

Among the first questions zmconfig.pl asks you are to do with the database and the next thing you should do is create it and the associated database users. You may notice that there are two sets of users and passwords. This is because the streaming server and utility binaries require only read access to the database so you may wish to create both a full access user and a limited access user. You can of course set both to the full access user. The included schema (zmschema.sql) can be used to actually create the tables required. The database is usually called just 'zm'.

If you are a first time user the first run of zmconfig.pl will warn you about the missing database, you can ignore those errors this time. Once you've run it for the first time the schema file should have your desired database name in it so use it to create the database (see below). Once the database and permissions are set up rerun zmconfig.pl with the *-noi* option to get it to load the configuration into your new database.

If you are upgrading from a previous version you can use `zmalter-x.y.z.sql` to upgrade your database and make the necessary changes where `x.y.z` identifies the version of ZoneMinder you had installed previously. So if you are going from version 0.9.7 to version 0.9.11 you would run the scripts for all intervening versions to get to the current one, i.e. `zmalter-0.9.7.sql`, `zmalter-0.9.8.sql`, `zmalter-0.9.9.sql` and `zmalter0.9.10.sql`. Additionally from version 1.19.5 you can use the `zmupdate.pl` script to do all the updating for you, this will include all the database changes plus any other tweaks necessary to upgrade to the latest version. To use it you should type

```
zmupdate.pl --version=<previous version> [--user=<db user> --pass=<db password>]
```

where 'previous version' relates to the version of ZM you are upgrading from, 1.19.4 for example. All updates from that versions onwards will be applied, however `zmupdate.pl` will only work with upgrades from 1.19.0 onwards. The 'user' and 'pass' options allow you to specify a database user and password with sufficient privilege to 'alter' the structure of the database. This is not necessarily the database user you use for ZoneMinder itself as specified in the section below.

For a new installation the simplest way to create your database and users is as follows,

```
mysql mysql < zmschema.sql
```

```
mysql mysql
```

```
grant select,insert,update,delete on <your database name>.* to <username>'@localhost identified by <password>;
```

```
quit
```

```
mysqladmin reload
```

You may need to supply a username and password to the `mysql` commands in the first place to give yourself sufficient privileges to perform the required commands. If you want to host your database on a different machine than that which ZoneMinder is running on then use the hostname of the remote machine instead of `localhost`.

Then just type '`make`' and off you go.

## 4. Installation

Once the build has completed you should have several shiny new binaries. I will now briefly describe what each of them does.

**zmc** - This is the ZoneMinder Capture daemon. This binary's job is to sit on a video device and suck frames off it as fast as possible, this should run at more or less constant speed.

**zma** - This is the ZoneMinder Analysis daemon. This is the component that goes through the captured frames and checks them for motion which might generate an alarm or event. It generally keeps up with the Capture daemon but if very busy may skip some frames to prevent it falling behind.

**zmf** - This is the ZoneMinder Frame daemon. This is an optional daemon that can run in concert with the Analysis daemon and whose function it is to actually write captured frames to disk. This frees up the Analysis daemon to do more analysis (!) and so keep up with the Capture daemon better. If it isn't running or dies then the Analysis daemon just writes them itself.

**zms** - This is the ZoneMinder Streaming server. The web interface connects with this to get real-time or historical streamed images. It runs only when a live monitor stream or event stream is actually being viewed and dies when the event finishes or the associate web page is closed. If you find you have several zms processes running when nothing is being viewed then it is likely you need a patch for apache (see the Troubleshooting section).

**zmu** - This is the ZoneMinder Utility. It's basically a handy command line interface to several useful functions. It's not really meant to be used by anyone except the web page (there's only limited 'help' in it so far) but can be if necessary, especially for debugging video problems.

**zmfix** - This is a small binary that exists only to ensure that the video device files can be read by the main capture daemons. It is often the case that these device files are set to be accessible by root only on boot. This binary runs setuid and ensures that they have appropriate permissions. This is not a daemon and runs only on system start and then exits.

As well as this there are the web PHP files in the web directory and some perl scripts in the scripts directory. These scripts all have some configuration at the top of the files which should be viewed and amended if necessary and are as follows.

**zmpkg.pl** - This is the ZoneMinder Package Control script. This is used by the web interface and service scripts to control the execution of the system as a whole.

**zmdc.pl** - This is the ZoneMinder Daemon Control script. This is used by the web interface and the zmpkg.pl script to control and maintain the execution of the capture and analysis daemons, amongst others. You should not need to run this script yourself.

**zmfilter.pl** - This script controls the execution of saved filters and will be started and stopped by the web interface based on whether there are filters that have been defined to be autonomous. This script is also responsible for the automatic uploading of events to a 3rd party server.

**zmaudit.pl** - This script is used to check the consistency of the event file system and database. It can delete orphaned events, i.e. ones that appear in one location and not the other as well as checking that all the various event related tables are in line. It can be run interactively or in batch mode either from the command line or a cron job or similar. In the zmconfig.pl there is an option to specify fast event deletes where the web interface only deletes the event entry from the database itself. If this is set then it is this script that tidies up the rest.

**zmwatch.pl** - This is a simple script purely designed to keep an eye on the capture daemons and restart them if they lockup. It has been known for sync problems in the video drivers to cause this so this script makes sure that nothing important gets missed.

**zmupdate.pl** - Currently this script is responsible for checking whether a new version of ZoneMinder is available and other miscellaneous actions related to upgrades and migrations. It is also intended to be a 'one stop shop' for any upgrades and will execute everything necessary to update your installation to a new version.



**zmx10.pl** - This is an optional script that can be used to initiate and monitor X10 Home Automation style events and interface with an alarm system either by the generation of X10 signals on ZoneMinder events or by initiating ZoneMinder monitoring and capture on receipt of X10 signals from elsewhere, for instance the triggering of an X10 PIR. For example I have several cameras that don't do motion detection until I arm my alarm system whereupon they switch to active mode when an X10 signal is generated by the alarm system and received by ZoneMinder.

**zmtrigger.pl** - This is an optional script that is a more generic solution to external triggering of alarms. It can handle external connections via either internet socket, unix socket or file/device interfaces. Currently you have to configure these using the 'sources' array in the script itself though ultimately I would expect they would be databased. The format of triggers used by zmtrigger.pl is as follows "`<id><action><score><cause><text>`" where

- 'id' is the id number or name of the ZM monitor
- 'action' is 'on','off' or 'cancel' where 'on' forces an alarm condition on, 'off' forces an alarm condition off and 'cancel' negates the previous 'on' or 'off'. Ordinarily you would use 'on' and 'cancel', 'off' would tend to be used to suppress motion based events. Additionally 'on' and 'off' can take an additional time offset, e.g. on+20 which automatically 'cancel's the previous action after that number of seconds.
- 'score' is the score given to the alarm, usually to indicate it's importance. For 'on' triggers it should be non-zero, otherwise it should be zero.
- 'cause' is a 32 char max string indicating the reason for, or source of the alarm e.g. 'Relay 1 open'. Ignored for 'off' or 'cancel' messages
- 'text' is a 256 char max additional info field, currently not used for anything though anything passed in here will get saved. Ignored for 'off' or 'cancel' messages.

Note that multiple messages can be sent at once and should be LF or CRLF delimited. Without customisation zmtrigger.pl is of little use, but if you configure it correctly it can help integrate other systems to generate external triggers that will cause ZoneMinder events.

**zmcontrol-\*.pl** – These are a set of example scripts which can be used to control Pan/Tilt/Zoom class cameras. Each script converts a set of standard parameters used for camera control into the actual protocol commands sent to the camera. If you are using a camera control protocol that is not in the shipped list then you will have to create a similar script though it can be created entirely separately from ZoneMinder and does not need to be named as these scripts are. Although the scripts are used to action commands originated from the web interface they can also be used directly or from other programs or scripts, for instance to implement periodic scanning to different presets.

**zmtrack.pl** – This script is used to manage the experimental motion tracking feature. It is responsible for detecting that an alarm is taking place and moving the camera to point to the alarmed location, and then subsequently returning it to a defined standby location. As well as moving the camera it also controls when motion detection is suspended and restored so that the action of the camera tracking does not trigger endless further alarms which are not justified.

**zm** - This is the (optional) ZoneMinder init script, see below for details.

Finally, check `zm_config.php` in the web directory and amend any configuration necessary in there. Most will have already been done by the configuration utilities but some scripts have a 'VERBOSE' flag you can set to get more debug out.

At this stage typing 'make install' will install everything to the desired locations, you may wish to su to root first though. The installation routine will copy the binaries and scripts to your chosen install location, usually `/usr/local/bin` and then move zms to your cgi-bin area. It will then copy the web files to your chosen directory and ensure they have the right permissions. Finally it tries to link `zm.php` to `index.php` but will not overwrite an existing file if it already exists.

The 'zm' script does not get installed automatically as it is not necessary for the operation of the ZoneMinder setup per se and is not necessarily supported for distributions other than those from the RedHat or Fedora families. However if you want to ensure that the ZoneMinder daemons are started on reboot etc copy it to your init.d directory, usually something like `/etc/rc.d/init.d` and then add it by doing

*chkconfig --add zm*

or similar command for your distribution. ZoneMinder will then start up when your machine reboots and can be controlled (by the root user) by doing *'service zm start'* or *'service zm stop'* etc. You may need to use the *'—levels'* parameter to *chkconfig* to ensure that ZoneMinder is started when you need it to.

Now start your web browser, point it at your *zm.php* and off you go.

## 5. Installing from RPM

Installing from the RPM is Distribution specific so make sure you download the correct RPM for the distribution that you are using.

All documents including this README are installed to the systems default document folder.

Fedora Core: `/usr/share/doc/zm-{version number}`

Mandrake:

The packaged version of Zone Minder installs all binaries to `/usr/lib/zm` including the web pages. So don't worry when you do not see any files installed to the root directory for your web server. The web pages for Apache are aliased by `zm.conf` in the `apache/conf.d` directory which vary depending on your distribution:

Fedora Core: `/etc/httpd/conf.d/zm.conf`

Mandrake:

The Configuration file for setting up the database is located at `/etc/zm.conf` and will need to be edited to add the user and password that you want Zone Minder to use. After you have installed the Zone Minder package this will be the first thing you want to do. So use your favourite editor and add in the user name and password you want Zone Minder to use. You can also change the database name if you would like.

```
vi /etc/zm.conf
```

Start the `mysqld` service so you can build the database

```
service mysqld start
```

Then run `zminit` to create the database

```
/usr/lib/zm/bin/zminit
```

The user and password that `zminit` asks for are for the database only. For the user enter `root` and leave the password blank (unless of course you changed the password). You should see some information showing that it has created the database and no errors.

Set the run levels for the services that Zone Minder requires. I like to set the run levels to 3 and 5 with the following command:

```
chkconfig --levels 35 mysqld on
```

```
chkconfig --levels 35 httpd on
```

Now start the web server and Zone Minder:

```
service httpd start
```

```
service zm start
```

You should now be able to access the Zone Minder console through the web browser <http://localhost/zm>

Log files will be located in `/var/log/zm`

Events are located at `/var/lib/zm`

## 6. Tutorial

What you see now (and subsequently) depends on whether you chose to run ZoneMinder in authenticated mode or not. This is an option that lets you specify whether anyone that goes to the ZoneMinder web pages must authenticate themselves in order to be given permissions to perform certain tasks. If you chose this mode then you will need to log in here. By default a fully privileged user 'admin' has been created with a password also of 'admin'. You should change this password as soon as possible.

Once you've logged in, or if you are running in un-authenticated mode, you will now see the ZoneMinder Console window. This will resize itself to avoid being too intrusive on your desktop. Along the top there are several informational entries like the time of the last update and the current server load. There will also be an indication of the system state which will probably say 'stopped' to start with. This is a link that you can click on to control the ZoneMinder system as a whole. Below that are various other links including one detailing the current user (in authenticated mode only) and one allowing you to configure your bandwidth. This last one enables you to optimise your settings depending on where you are, the actual values relating to this are defined in the options. If you are using a browser on the same machine or network then choose high, over a cable or DSL link maybe choose medium and over a dialup choose low. You can experiment to see which is best. This setting is retained on a per machine basis with a persistent cookie. Also on this line are a number of other links that will be covered below.

Please bear in mind that from here on the descriptions of the web pages are based on what you will see if you are running as a fully authenticated user. If you are running in un-authenticated mode or as a less privileged user then some elements may not be shown or will be disabled.

### 6.1. Defining Monitors

To use ZoneMinder properly you need to define at least one Monitor. Essentially, a monitor is associated with a camera and can continually check it for motion detection and such like. So, next click 'Add New Monitor' to bring up the dialog. You will see a bunch of things you have to fill in.

To help you get started on the video configuration the best thing is to use a tool like 'xawtv' (<http://bytesex.org/xawtv/>) to get a picture you're happy with, and to check your camera works. Then run 'zmu -d <device\_no> -q -v' to get a dump of the settings (note, you will have to additionally supply a username and password to zmu if you are running in authenticated mode). You can then enter these values into the video related options of the monitor configuration panel. The 'device\_no' referred to here is a number corresponding to the digit at the end of your device file, so /dev/video0 has a 'device\_no' of 0 etc. If 'zmu' gives you an error related to permissions run 'zmfix -a' to make sure you can access all the video devices.

The options are divided into a set of tabs to make it easier to edit. You do not have to 'save' to change to different tab so you can make all the changes you require and then click 'Save' at the end. The individual option are explained in a little more detail below,

#### 'Monitor' Tab

**Name** – The name for your monitor. This should be made up of alphanumeric characters (a-z,A-Z,0-9) and hyphen (-) and underscore(\_) only. Whitespace is not allowed.

**Function** – This essentially defines what the monitor is doing. This can be one of the following;

- 'None' – The monitor is currently disabled and no streams can be viewed or events generated.
- 'Monitor' – The monitor will only stream feeds but no image analysis is done and so no alarms or events will be generated,
- 'Modect' – or MOtion DEteCTtion. All captured images will be analysed and events generated where motion is detected.

- 'Record' – In this case continuous events of a fixed length are generated regardless of motion which is analogous to a convention time-lapse video recorder. No motion detection takes place in this mode.
- 'Mocord' – This is a hybrid of Modect and Record and results in both fixed length events being recorded and also any motion being highlighted within those events.
- 'Nodect' – or No DEteCTtion. This is a special mode designed to be used with external triggers. In Nodect no motion detection takes place but events are recorded if external triggers require it.

Generally speaking it is best to choose 'Monitor' as an initial setting here..

**Section Length** – This specifies the length (in seconds) of any fixed length events produced when the monitor function is 'Record' or 'Mocord'. Otherwise it is ignored. This should not be so long that events are difficult to navigate nor so short that too many events are generated. A length of between 300 and 900 seconds I recommended.

**Frame Skip** – This setting also applies only to the 'Record' or 'Mocord' functions and specifies how many frames should be skipped in the recorded events. The default setting of zero results in every captured frame being saved, whereas one would mean that one frame is skipped between each saved one, two means that two frames are skipped between each saved one etc. An alternate way of thinking is that one in every 'Frame Skip + 1' frames is saved. The point of this is to ensure that saved events do not take up too much space unnecessarily whilst still allowing the camera to capture at a fairly high frame rate. The alternate approach is to limit the capture frame rate which will obviously affect the rate at which frames are saved.

**Run Mode** – Two choices are available here. 'Continuous' is the usual setting and means that the monitor is expected to be performing the function selected above at all times and should one or more of the daemons fail or not be running it will be automatically restarted. By contrast 'Triggered' means that the decision about whether the daemons should actually be active is devolved to an external triggering mechanism.

**Triggers** – This small section lets you select which triggers will apply if the run mode has been set to 'triggered' above. The most common trigger is X10 and this will appear here if you indicated that your system supported it during installation. Only X10 is supported as a shipped trigger with ZoneMinder at present but it is expected that other triggers will become available as necessary. You can also just use 'cron' jobs or other mechanisms to actually control the camera and keep them completely outside of the ZoneMinder settings.

**Source Type** – This determines whether the camera is a local one attached to a physical video or USB port on your machine or a remote network camera or similar. Choosing one or the other affects which set of options are shown in the next tab.

#### 'Source' Tab (local device)

**Device Number/Channel** – For a local camera enter the device number that your camera is attached to. If it is /dev/video0 enter '0' etc. Some video devices, e.g. BTTV cards support multiple cameras on one device so in this case enter the channel number in the Channel box or leave it at zero if you're using a USB camera or one with just one channel.

**Device Format** – For a local camera enter the video format of the video stream. This is defined in various system files (e.g. /usr/include/linux/videodev.h) but the two most common are 0 for PAL and 1 for NTSC.

**Capture Palette** – Finally for the video part of the configuration enter the colour depth. ZoneMinder supports a handful of the most common palettes, so choose one here. If in doubt try grey first, and then 24 bit colour. If neither of these work very well then YUV420P or one of the others probably will. There is a slight performance penalty when using palettes other than grey or 24 bit colour as an internal conversion is involved. These other formats are intended to be supported natively in a future version but for now if you have the choice choose one of grey or 24 bit colour.

**Capture Width/Height** – The dimensions of the video stream your camera will supply. If your camera supports several just enter the one you'll want to use for this application, you can always change it later. However I would recommend starting with no larger than 320x240 or 352x288 and then perhaps increasing and seeing how performance is affected. This size should be adequate in most cases. Some cameras are quite choosy about the sizes you can use here so unusual sizes such as 197x333 should be avoided initially.

**Orientation** – If your camera is mounted upside down or at right angles you can use this field to specify a rotation that is applied to the image as it is captured. This incurs an additional processing overhead so if possible it is better to mount your camera the right way round if you can. If not set the orientation here. If you choose one of the rotation options remember to switch the height and width fields so that they apply, e.g. if your camera captures at 352x288 and you choose 'Rotate Right' here then set the height to be 352 and width to be 288.

#### 'Source' Tab (remote device)

**Remote Host/Port/Path** – For remote cameras use these fields to enter the full URL of the camera. Basically if your camera is at <http://camserver.home.net:8192/cameras/camera1.jpg> then these fields will be camserver.home.net, 8192 and /cameras/camera1.jpg respectively. Leave the port at 80 if there is no special port required. If you require authentication to access your camera then add this onto the host name in the form <username>:<password>@<hostname>.com.

**Remote Image Colours** – Specify the amount of colours in the captured image. Unlike with local cameras changing this has no controlling effect on the remote camera itself so ensure that your camera is actually capturing to this palette beforehand.

**Capture Width/Height** – As per local devices.

**Orientation** – As per local devices.

#### 'Timestamp' Tab

**Timestamp Label Format** – This relates to the timestamp that is applied to each frame. It is a 'sprintf' style string. It is actually passed through sprintf and then through printf to add the monitor name so a format of '%s - %y/%m/%d %H:%M:%S' (note the double % at the beginning) would be recommended though you can modify it if necessary. If you don't want a timestamp or have a camera that puts one on itself then leave this field blank.

**Timestamp Label X/Y** – The X and Y values determine where to put the timestamp. A value of 0 for the X value will put it on the left side of the image and a Y value of 0 will place it at the top of the image. To place the timestamp at the bottom of the image use a value eight less than the image height.

#### 'Buffers' Tab

**Image Buffer Size** – This option determines how many frames are held in the ring buffer at any one time. The ring buffer is the storage space where the last 'n' images are kept, ready to be resurrected on an alarm or just kept waiting to be analysed. It can be any value you like with a couple of provisos, (see next options). However it is stored in shared memory and making it too large especially for large images with a high colour depth can use a lot of memory. A value of no more than 50 is usually ok. If you find that your system will not let you use the value you want it is probably because your system has an arbitrary limit on the size of shared memory that may be used even though you may have plenty of free memory available. This limit is usually fairly easy to change, see the Troubleshooting section for details.

**Warm-up Frames** – This specifies how many frames the analysis daemon should process but not examine when it starts. This allows it to generate an accurate reference image from a series of images before looking too carefully for any changes. I use a value of 25 here, too high and it will take a long time to start, too low and you will get false alarms when the analysis daemon starts up.

**Pre/Post Event Image Buffer** – These options determine how many frames from before and after an event should be preserved with it. This allows you to view what happened immediately

prior and subsequent to the event. A value of 10 for both of these will get you started but if you get a lot of short events and would prefer them to run together to form fewer longer ones then increase the Post Event buffer size. The pre-event buffer is a true buffer and should not really exceed half the ring buffer size. However the post-event buffer is just a count that is applied to captured frames and so can be managed more flexibly. You should also bear in mind the frame rate of the camera when choosing these values. For instance a network camera capturing at 1FPS will give you 10 seconds before and after each event if you chose 10 here. This may well be too much and pad out events more than necessary. However a fast video card may capture at 25FPS and you will want to ensure that this setting enables you to view a reasonable time frame pre and post event.

**Alarm Frame Count** – This option allows you to specify how many consecutive alarm frames must occur before an alarm event is generated. The usual, and default, value is 1 which implies that any alarm frame will cause or participate in an event. You can enter any value up to 16 here to eliminate bogus events caused perhaps by screen flickers or other transients. Values over 3 or 4 are unlikely to be useful however. Please note that if you have statistics recording enabled then currently statistics are not recorded for the first 'Alarm Frame Count'-1 frames of an event. So if you set this value to 5 then the first 4 frames will be missing statistics whereas the more usual value of 1 will ensure that all alarm frames have statistics recorded.

### 'Misc' Tab

**Event Prefix** – By default events are named 'Event-<event id>', however you are free to rename them individually as you wish. This option lets you modify the event prefix, the 'Event-' part, to be a value of your choice so that events are named differently as they are generated. This allows you to name events according to which monitor generated them.

**Maximum FPS** – On some occasions you may have one or more cameras capable of high capture rates but find that you generally do not require this performance at all times and would prefer to lighten the load on your server. This option permits you to limit the maximum capture rate to a specified value. This may allow you to have more cameras supported on your system by reducing the CPU load or to allocate video bandwidth unevenly between cameras sharing the same video device. This value is only a rough guide and the lower the value you set the less close the actual FPS may approach it especially on shared devices where it can be difficult to synchronise two or more different capture rates precisely. There is a global configuration option that allows you to turn this limiting off in the event of an alarm.

**FPS Report Interval** – How often the current performance in terms of Frames Per Second is output to the system log. Not used in any functional way so set it to maybe 1000 for now. If you watch /var/log/messages (normally) you will see this value being emitted at the frequency you specify both for video capture and processing.

**Reference Image Blend %age** – Each analysed image in ZoneMinder is a composite of previous images and is formed by applying the current image as a certain percentage of the previous reference image. Thus, if we entered the value of 10 here, each image's part in the reference image will diminish by a factor of 0.9 each time round. So a typical reference image will be 10% the previous image, 9% the one before that and then 8.1%, 7.2%, 6.5% and so on of the rest of the way. An image will effectively vanish around 25 images later than when it was added. This blend value is what is specified here and if higher will make slower progressing events less detectable as the reference image would change more quickly. Similarly events will be deemed to be over much sooner as the reference image adapts to the new images more quickly. In signal processing terms the higher this value the steeper the event attack and decay of the signal. It depends on your particular requirements what the appropriate value would be for you but start with 10 here and adjust it (usually down) later if necessary.

### 'Control' Tab

**Note:** This tab and its options will only appear if you have selected the **ZM\_OPT\_CONTROL** option to indicated that your system contains cameras which are able to be controlled via Pan/Tilt/Zoom or other mechanisms. See the Camera Control section elsewhere in this document for further details on camera control protocols and methods.

**Controllable** – Check this box to indicate your camera can be controlled.

**Control Type** – Select the control type that is appropriate for your camera. ZoneMinder ships with a small number of predefined control protocols which will work with some cameras without modification but which may have to be amended to function with others. Choose the edit link to create new control types or to edit the existing ones.

**Control Device** – This is the device that is used to control your camera. This will normally be a serial or similar port. If your camera is a network camera, you will generally not need to specify a control device.

**Control Address** – This is the address of your camera. Some control protocols require that each camera is identified by a particular, usually numeric id. If your camera uses addressing then enter the id of your camera here. If your camera is a network camera then you will usually need to enter the hostname or IP address of it here. This is ordinarily the same as that given for the camera itself.

**Track Motion** – This and the following four options are used with the experimental motion function. This will only work if your camera supports mapped movement modes where a point on an image can be mapped to a control command. This is generally most common on network cameras but can be replicated to some degree on other cameras that support relative movement modes. See the Camera Control section for more details. Check this box to enable motion tracking.

**Track Delay** – This is the number of seconds to suspend motion detection for following any movement that the camera may make to track motion.

**Return Location** – If your camera supports a 'home' position or presets you can choose which preset the camera should return to after tracking motion.

**Return Delay** – This is the delay, in seconds, once motion has stopped being detected, before the camera returns to any defined return location.

## 'X10' Tab

**Note: This tab and its options will only appear if you have indicated that your system supports the X10 home automation protocol during initial system configuration.**

**X10 Activation String** - The contents of this field determine when a monitor starts and/or stops being active when running in 'Triggered' mode and with X10 triggers. The format of this string is as follows,

n : If you simply enter a number then the monitor will be activated when an X10 ON signal for that unit code is detected and will be deactivated when an OFF signal is detected.

!n : This inverts the previous mode, e.g. !5 means that the monitor is activated when an OFF signal for unit code 5 is detected and deactivated by an ON.

n+ : Entering a unit code followed by + means that the monitor is activated on receipt of a ON signal for that unit code but will ignore the OFF signal and as such will not be deactivated by this instruction. If you prepend a '!' as per the previous definition it similarly inverts the mode, i.e. the ON signal deactivates the monitor.

n+<seconds> : As per the previous mode except that the monitor will deactivate itself after the given number of seconds.

n- : Entering a unit code followed by - means that the monitor is deactivated on receipt of a OFF signal for that unit code but will ignore the ON signal and as such will not be activated by this instruction. If you prepend a '!' as per the previous definition it similarly inverts the mode, i.e. the OFF signal activates the monitor.

n-<seconds> : As per the previous mode except that the monitor will activate itself after the given number of seconds.



You can also combine several of these expressions to by separating them with a comma to create multiple circumstances of activation. However for now leave this blank.

**X10 Input Alarm String** - This has the same format as the previous field but instead of activating the monitor with will cause a forced alarm to be generated and an event recorded if the monitor is Active. The same definition as above applies except that for activated read alarmed and for deactivated read unalarmed(!). Again leave this blank for now.

**X10 Output Alarm String** - This X10 string also has the same format as the two above options. However it works in a slightly different way. Instead of ZoneMinder reacting to X10 events this option controls how ZoneMinder emits X10 signals when the current monitor goes into or comes out of the alarm state. Thus just entering a number will cause the ON signal for that unit code to be sent when going into alarm state and the OFF signal when coming out of alarm state. Similarly 7+30 will send the unit code 7 ON signal when going into alarm state and the OFF signal 30 seconds later regardless of state. The combination of the X10 instruction allows ZoneMinder to react intelligently to, and also assume control of, other devices when necessary. However the indiscriminate use of the Input Alarm and Output Alarm signals can cause some horrendous race conditions such as a light going on in response to an alarm which then causes an alarm itself and so on. Thus some circumspection is required here. Leave this blank for now anyway.

Finally, click 'Save' to add your monitor.

On the main console listing you will now see your monitor and some of its vital statistics. Most columns are also links and you get to other functions of ZoneMinder by choosing the appropriate one. Describing them left to right, they are as follows.

The first column is the Id, clicking on this gives you the opportunity to edit any of the settings you have just defined your monitor to have.

The next column is the Name column, clicking on this will give you the watch window where you can view a live feed from your camera along with recent events. This is described more fully below.

Following that are the Function and Source columns, which may be represented in various colours. Initially both will be showing red. This means that that monitor is not configured for any function and as a consequence has no zmc (capture) daemon running on it. If it were orange it would mean that a zmc daemon was running but no zma (analysis) daemon and green means both are running. In our case it is red because we defined the Monitor to have a Function of None so no daemons are required. To get the daemons up and running you can either click on the source listed in the Source column and edit the monitor properties or click on the Function listed and change it to 'Monitor', which will ensure that one or more appropriate daemons are started automatically.

Having a device status of red or orange does not necessarily constitute an error if you have deliberately disabled a monitor or have just put it into Passive mode.

If you have several cameras (and thus monitors) on a device the device status colour reflects all of them for the capture daemon. So if just one monitor is active then the daemon is active for both even if all the other monitors are switched off.

Once you have changed the function of your monitor, the main console window will be updated to reflect this change. If your device status does not go green then check your system and web server logs to see if it's something obvious.

You can now add further monitors if you have cameras set up to support them. Once you have one or more monitors you may notice the '<n> Monitors' title becomes a link. Clicking on this link will open up a window which allows you to assign your monitors to groups. These let you select certain monitors to view. For instance you may only wish to view outdoor monitors while indoors. You can also choose to view all of them. If you choose a group then your selection will be remembered via a cookie and will be used until you change it. You can call your groups anything you like, though 'Mobile' has a special meaning (see Mobile Devices below). There may also be a 'Cycle' link which allows you to cycle through a shot from each of your monitors (in the selected group unless they are switched off) and get a streamed or still image from each in turn. Similarly if you see a link titled 'Montage' it will allow you view all your active enabled cameras (in the selected group)

simultaneously. Be aware however that this can consume large amounts of bandwidth and CPU so should not be used continuously unless you have resource to burn.

## 6.2. Defining Zones

The next important thing to do with a new monitor is set up Zones for it to use. By default you'll already have one created for you when you created your monitor but you might want to modify it or add others. Click on the Zones column for your monitor and you should see a small popup window appear which contains an image from your camera overlain with a stippled pattern representing your zone. In the default case this will cover the whole image and will be red. Beneath that will be a table containing a listing of your zones. Clicking on either the relevant bit of the image or on the Id or Name in the table will bring up another window where you can edit the particulars for your Zones. As you can see there are quite a few, so now is a good time to go through them. The options are as follows.

**Name** – This is just a label to identify the zone by. You can change this to be more representative if you like, though it isn't used much except for logging and debugging.

**Type** - This is one of the more important concepts in ZoneMinder and there are five to choose from.

Active : This is the zone type you'll use most often, and which will be set for your default zone. This means that this zone will trigger an alarm on any events that occur within it that meet the selection criteria.

Inclusive : This zone type can be used for any zones that you want to trigger an alarm only if at least one other Active zone has already triggered one. This might be for example to cover an area of the image like a plant or tree which moves a lot and which would trigger lots of alarms. Perhaps this is behind an area you'd like to monitor though, in this case you'd create an active zone covering the non-moving parts and an inclusive zone covering the tree perhaps with less sensitive detection settings also. If something triggered an alarm in the Active zone and also in the Inclusive zone they would both be registered and the resulting alarm would be that much bigger than if you had blanked it out altogether.

Exclusive : The next zone Type is Exclusive. This means that alarms will only be triggered in this zone if no alarms have already been triggered in Active zones. This is the most specialised of the zone types and you may never use it but in its place it is very useful. For instance in the camera covering my garden I keep watch for a hedgehog that visits most nights and scoffs the food out of my cats bowls. By creating a sensitive Exclusive zone in that area I can ensure that a hedgehog alarm will only trigger if there is activity in that small area. If something much bigger occurs, like someone walking by it will trigger a regular alarm and not one from the Exclusive zone. Thus I can ensure I get alarms for big events and also special small events but not the noise in between.

Preclusive : This zone type is relatively recent. It is called a Preclusive zone because if it is triggered it actually precludes an alarm being generated for that image frame. So motion or other changes that occur in a Preclusive zone will have the effect of ensuring that no alarm occurs at all. The application for this zone type is primarily as a shortcut for detecting general large-scale lighting or other changes. Generally this may be achieved by limiting the maximum number of alarm pixels or other measure in an Active zone. However in some cases that zone may cover an area where the area of variable illumination occurs in different places as the sun and/or shadows move and it thus may be difficult to come up with general values. Additionally, if the sun comes out rapidly then although the initial change may be ignored in this way as the reference image catches up an alarm may ultimately be triggered as the image becomes less different. Using one or more Preclusive zones offers a different approach. Preclusive zones are designed to be fairly small, even just a few pixels across, with quite low alarm thresholds. They should be situated in areas of the image that are less likely to have motion occur such as high on a wall or in a corner. Should a general illumination change occur they would be triggered at least as early as any Active zones and prevent any other zones from generating an alarm. Obviously careful placement is required to ensure that they do not cancel any genuine alarms or that they are not so close together that any motion just hops from one Preclusive zone to another. As always, the best way is to experiment a little and see what works for you.

Inactive : This final zone type is the opposite of Active. In this zone type no alarms will ever be reported. You can create an Inactive zone to cover any areas in which nothing notable will ever happen or where you get constant false alarms that don't relate to what you are trying to monitor. An Inactive zone can overlay other zone types and will be processed first.

I mentioned above that Inactive zones may be overlaid on other zones to blank out areas however as a general principle you should try and make zones abut each other as much as possible and do not overlap. This helps avoid repeated duplicate processing of the same area. For instance an Inclusive zone overlaying an Active zone when all other settings are the same will always trigger when the Active zone does which somewhat defeats the object of the exercise. One exception to this is Preclusive zones. These may be situated within Active areas are they are processed first and if small may actually save processing time by preventing full analysis of the image.

**Units** - This setting which details whether certain of the following settings are in Pixels or Percent, where 'Percent' refers to a percentage area of the zone itself. In general 'Pixels' is more precise whereas percentages are easier to use to start with or if you change image sizes frequently. If you change this setting all appropriate values below are redisplayed in the correct context. A good tip would be to initially enter the settings in Percent and then change to Pixels and refine any gaps. Repeated flipping between the settings will cause rounding errors, as ZoneMinder in general is not at home to Mr Floating Point for reasons of performance. Note, the sense of the percentage values changed in version 1.19.0. Prior to that percentages referred to the area of the image as a whole, whereas it now only refers to the area of the zone. This makes trying to work out necessary sizes rather easier.

**Min/Maximum X/Y** - Following the units the next four settings define the bounds of the Zone in the monitor frame and are self-explanatory with the exception of the fact that the minima are at the top left of the frame and the maxima are at the bottom right rather than in a Cartesian style.

**Alarm Colour** - The option after that allows you to specify what colour you'd like any alarms this zone generates to be highlighted on images, pick anything you like that will show up against your normal image background. This option is irrelevant for Preclusive and Inactive zones and will be disabled For Inactive zones all subsequent options are likewise disabled.

**Alarm Check Method** –This is a new addition to Zone definitions. It allows you to specify the nature of the alarm checking that will take place, and more specifically what tests are applied to determine whether a frame represents an alarm or not. The three options are 'AlarmPixels', 'FilteredPixels' and 'Blobs' and depending on which option is chosen some of the following other settings may become unavailable. The first of these indicates that only a count of individual alarmed pixels should be used to determine the state of a image, the second indicate that the pixels should be filtered to remove isolated pixels (see below) before being counted, and the third uses a more sophisticated analysis which is designed to aggregate alarmed pixels into continuous groups, or 'blobs'. Blob analysis is the method ZoneMinder has always used previously (before it became optional) and so this is the default. However this method takes slightly longer and so if you find that one of the other methods works just as well for you and you wish to maximise performance you can opt for that instead. Some of the more useful alarm related features such as highlighted analysis images are only available with the 'Blob' setting.

**Min/Maximum Pixel Threshold** – These setting are used to define limits for the difference in value between a pixel and its predecessor in the reference image. For greyscale images this is simple but for colour images the colours are averaged first, originally this used an RMS (root mean squared) algorithm but calculating square roots mugs performance and does not seem to improve detection. Using an average does means that subtle colour changes without any brightness change may go undetected but this is not the normal circumstance. There is also the option to use a more sophisticated integer algorithm to calculate a Y (or brightness) value from the colours themselves.

**Min/Maximum Alarmed Area** - The following two settings define the minimum and maximum number of pixels that exceed this threshold that would cause an alarm. If the units are Percent this (and following options) refers to the percentage of the frame and not the zone, this is so these values can be related between zones. The minimum value must be matched or exceeded for an alarm to be generated whereas the maximum must not be exceeded or the

alarm will be cancelled. This is to allow for sudden changes such as lights coming on etc, which you may wish to disregard. In general a value of zero for any of these settings causes that value to be ignored, so you can safely set a maximum to zero and it will not be used. The use of just a number of pixels is however a very brute force method of detection as many small events dispersed widely are not distinguished from a compact one.

**Filter Width/Height** – To improve detection of valid event ZoneMinder applies several other functions to the data to improve its ability to distinguish interesting signals from uninteresting noise. The first of these is a filter that removes any pixels that do not participate in a contiguous block of pixels above a certain size. These options are always expressed in pixels and should be fairly small, and an odd number, three or five is a good value to choose initially. Application of this filter removes any tiny or discontinuous pixels that don't form part of a discrete block.

**Min/Maximum Filtered Area** – These are two additional bounds that specify the limits of pixels that would cause an alarm after this filtering process. As the filtering process can only remove alarmed pixels it makes no sense for the Minimum and Maximum Filtered Area to be larger than the equivalent Alarmed Area and in general they should be smaller or the same.

**Min/Maximum Blob Area** - The next step in the analysis phase is the collation of any remaining alarmed areas into contiguous blobs. This process parses the image and forms any pixels that adjoin other alarmed pixels into one or more larger blobs. These blobs may be any shape and can be as large as the zone itself or as small as the filtered size. The Minimum and Maximum Blob Size settings allow you to define limits within which an alarm will be generated. Of these only the Minimum is likely to be very useful.

**Min/Maximum Blobs** - Finally the Minimum and Maximum Blobs settings specify the limits of the actual number of blobs detected. If an image change satisfies all these requirements it starts or continues an alarm event.

### **6.3. Viewing Monitors**

As this point you should have one or more Monitors running with one or more Zones each. Returning to the main Console window you will see your monitors listed once more. The columns not explored so far are the Monitor name, and various event totals for certain periods of time. Clicking on any of the event totals will bring up a variation on the same window but click on the Monitor name for now. On doing so up will pop another window which should be scaled to contain a heading, an image from your monitor, a status and a list of recent events if any have been generated. Depending on whether you are able to view a streamed image or not the image frame will either be this stream or a series of stills. You have the option to change from one to the other (if available) at the centre of the top heading. Also along the top are a handful of other links. These let you change the scale of the image stream, modify image settings (for local devices) or close the window. If you have cameras that can be controlled, a 'Control' link should also be preset which is described below.

The image should be self-explanatory but if it looks like garbage it is possible that the video configuration is wrong so look in your system error log and check for or report anything unusual. The centre of the window will have a tiny frame that just contains a status; this will be 'Idle', 'Alarm' or 'Alert' depending on the function of the Monitor and what's going on in the field of view. Idle means nothing is happening, Alarm means there is an alarm in progress and Alert means that an alarm has happened and the monitor is 'cooling down', if another alarm is generated in this time it will just become part of the same event. These indicators are colour coded in green, red and amber.

By default if you have minimised this window or opened other windows in front it will pop up to the front if it goes to Alarm state. This behaviour can be turned off in 'options' if required. You can also specify a sound file in the configuration, which will be played when an alarm occurs to alert you to the fact if you are not in front of your computer. This should be a short sound of only a couple of seconds ideally. Note that as the status is refreshed every few seconds it is possible for this not to alert you to every event that takes place, so you shouldn't rely on it for this purpose if you expect very brief events. Alternatively you can decrease the refresh interval for this window in the configuration though having too frequent refreshing may impact on performance.

Below the status is a list of recent events that have occurred, by default this is a listing of just the last 10 but clicking on 'All' will give you a full list and 'Archive' will take you to the event archive for

this monitor, more on this later. Clicking on any of the column headings will sort the events appropriately.

From here you can also delete events if you wish. The events themselves are listed with the event id, and event name (which you can change), the time that the event occurred, the length of the event including any preamble and postamble frames, the number of frames comprising the event with the number that actually contain an alarm in brackets and finally a score. This column lists the average score per alarm frame as well as the maximum score that any alarm frame had.

The score is an arbitrary value that essentially represents the percentage of pixels in the zone that are in blobs divided by the square root of the number of blobs and then divided by the size of the zone. This gives a nominal maximum of 100 for a zone and the totals for each zone are added together, Active zones scores are added unchanged, Inclusive zones are halved first and Exclusive zones are doubled. In reality values are likely to be much less than 100 but it does give a simple indication of how major the event was.

#### **6.4. Controlling Monitors**

If you have defined your system as having controllable monitors and you are looking at a monitor that is configured for control, then clicking on the 'Control' link along the top of the window will change the short event listing area to a control area. The capabilities you have defined earlier determine exactly what is displayed in this window. Generally you will have a Pan/Tilt control area along with one or subsidiary areas such as zoom or focus control to the side. If you have preset support then these will be near the bottom of the window. The normal method of controlling the monitor is by clicking on the appropriate graphics which then send a command via the control script to the camera itself. This may sometimes take a noticeable delay before the camera responds.

It is usually the case that the control arrows are sensitive to where you click on them. If you have a camera that allows different speeds to be used for panning or zooming etc then clicking near the point of the arrow will invoke the faster speed whilst clicking near the base of the arrow will be slower. If you have defined continuous motion then ongoing activities can be stopped by clicking on the area between the arrows, which will either be a graphic in the case of pan/tilt controls or a word in the case of zoom and focus controls etc. Certain control capabilities such as mapped motion allow direct control by clicking on the image itself when used in browsers which support streamed images directly. Used in this way you can just click on the area of the image that interests you and the camera will centre on that spot. You can also use direct image control for relative motion when the area of the image you click on defines the direction and the distance away from the centre of the image determines the speed. As it is not always very easy to estimate direction near the centre of the image, the active area does not start until a short distance away from the centre, resulting in a 'dead' zone in the middle of the image.

#### **6.5. Filtering Events**

The other columns on the main console window contain various event totals for your monitors over the last hour, day, week and month as well as a grand total and a total for events that you may have archived for safekeeping. Clicking on one of these totals or on the 'All' or 'Archive' links from the monitor window described above will present you with a new display. This is the full event window and contains a list of events selected according to a filter which will also pop up in its own window. Thus if you clicked on a 'day' total the filter will indicate that this is the period for which events are being filtered. The event listing window contains a similar listing to the recent events in the monitor window. The primary differences are that the frames and alarm frames and the score and maximum score are now broken out into their own columns, all of which can be sorted by clicking on the heading. Also this window will not refresh automatically, rather only on request. Other than that, you can choose to view events here or delete them as before.

The other window that appeared is a filter window. You can use this window to create your own filters or to modify existing ones. You can even save your favourite filters to re-use at a future date. Filtering itself is fairly simple; you first choose how many expressions you'd like your filter to contain. Changing this value will cause the window to redraw with a corresponding row for each expression. You then select what you want to filter on and how the expressions relate by choosing whether they are 'and' or 'or' relationships. For filters comprised of many expressions you will also get the option to bracket parts of the filter to ensure you can express it as desired. Then if you like choose how you want your results sorted and whether you want to limit the amount of events displayed.

There are several different elements to an event that you can filter on, some of which require further explanation. These are as follows, 'Date/Time' which must evaluate to a date and a time together, 'Date' and 'Time' which are variants which may only contain the relevant subsets of this, 'Weekday' which as expected is a day of the week. All of the preceding elements take a very flexible free format of dates and time based on the PHP strtotime function (<http://www.zend.com/manual/function.strptime.php>). This allows values such as 'last Wednesday' etc to be entered. I recommend acquainting yourself with this function to see what the allowed formats are. However automated filters are run in perl and so are parsed by the Date::Manip package. Not all date formats are available in both so if you are saved your filter to do automatic deletions or other tasks you should make sure that the date and time format you use is compatible with both methods. The safest type of format to use is '-3 day' or similar with easily parseable numbers and units are in English.

The other elements you can filter on are all fairly self explanatory except perhaps for 'Archived' which you can use to include or exclude Archived events. In general you'll probably do most filtering on un-archived events. There are also two elements, Disk Blocks and Disk Percent which don't directly relate to the events themselves but to the disk partition on which the events are stored. These allow you to specify an amount of disk usage either in blocks or in percentage as returned by the 'df' command. They relate to the amount of disk space used and not the amount left free. Once your filter is specified, clicking 'submit' will filter the events according to your specification. As the disk based elements are not event related directly if you create a filter and include the term 'DiskPercent > 95' then if your current disk usage is over that amount when you submit the filter then all events will be listed whereas if it is less then none at all will. As such the disk related terms will tend to be used mostly for automatic filters (see below). If you have created a filter you want to keep, you can name it and save it by clicking 'Save'.

If you do this then the subsequent dialog will also allow you specify whether you want this filter automatically applied in order to delete events or upload events via ftp to another server and mail notifications of events to one or more email accounts. In most cases you can specify your preferences for upload formats and email content during configuration time (make sure you type '?' to get help on options). Emails and messages (essentially small emails intended for mobile phones or pagers) have a variety of tokens that can be substituted for various details of the event that caused them. This includes links to the event view or the filter as well as the option of attaching images or videos to the email itself. See the included templates zmconfig\_eml.txt and zmconfig\_msg.txt for a fuller explanation of the availability and meaning of these tokens. Finally you can also specify a script which is run on each matched event. This script should be readable and executable by your web server user. It will get run once per event and the relative path to the directory containing the event in question. Normally this will be of the form <MonitorName>/<EventId> so from this path you can derive both the monitor name and event id and perform any action you wish. Note that arbitrary commands are not allowed to be specified in the filter, for security the only thing it may contain is the full path to an executable. What that contains is entirely up to you however.

Filtering is a powerful mechanism you can use to eliminate events that fit a certain pattern however in many cases modifying the zone settings will better address this. Where it really comes into its own is generally in applying time filters, so for instance events that happen during weekdays or at certain times of the day are highlighted, uploaded or deleted. Additionally using disk related terms in your filters means you can automatically create filters that delete the oldest events when your disk gets full. Be warned however that if you use this strategy then you should limit the returned results to the amount of events you want deleted in each pass until the disk usage is at an acceptable level. If you do not do this then the first pass when the disk usage is high will match, and then delete, all events unless you have used other criteria inside of limits.

## **6.6. Viewing Events**

From the monitor or filtered events listing you can now click on an event to view it in more detail. If you have streaming capability you will see a series of images that make up the event. You will also see a link to allow you to view the still images themselves. If you don't have streaming then you will be taken directly to this page. The images themselves are thumbnail size and depending on the configuration and bandwidth you have chosen will either be the full images scaled in your browser or actual scaled images. If it is the latter, if you have low bandwidth for example, it may take a few seconds to generate the images. If thumbnail images are required to be generated, they will be kept

and not re-generated in future. Once the images appear you can mouse over them to get the image sequence number and the image score.

You will notice for the first time that alarm images now contain an overlay outlining the blobs that represent the alarmed area. This outline is in the colour defined for that zone and lets you see what it was that caused the alarm. Clicking on one of the thumbnails will take you to a full size window where you can see the image in all its detail and scroll through the various images that make up the event. If you have the `ZM_RECORD_EVENT_STATS` option on, you will be able to click the 'Stats' link here and get some analysis of the cause of the event. Should you determine that you don't wish to keep the event, clicking on Delete will erase it from the database and file system. Returning to the event window, other options here are renaming the event to something more meaningful, refreshing the window to replay the event stream, deleting the event, switching between streamed and still versions of the event (if supported) and generating an MPEG video of the event (if supported).

These last two options require further explanation. Archiving an event means that it is kept to one side and not displayed in the normal event listings unless you specifically ask to view the archived events. This is useful for keeping events that you think may be important or just wish to protect. Once an event is archived it can be deleted or unarchived but you cannot accidentally delete it when viewing normal unarchived events.

The final option of generating an MPEG video is still somewhat experimental and its usefulness may vary. It can use either the Berkeley MPEG encoder or the faster and new ffmpeg encoder. Either of these will generate a short video, which will be downloaded to your browsing machine to view. Due to the relatively slow frame rate that ZoneMinder will capture at and the high minimum frame rate that the Berkeley encoder uses videos created by this method will be very quick. However when using the ffmpeg encoder, ZoneMinder will attempt to match the duration of the video with the duration of the event. This has the useful effect of making the video watchable and not too quick while having the unfortunate side effect of increasing file size and generation time. Ffmpeg in particular has a particularly rich set of options and you can specify during configuration which additional options you may wish to include to suit your preferences. In particular you may need to specify additional, or different, options if you are creating videos of events with particularly slow frame rates as some codecs only support certain ranges of frame rates. Details of these options can be found in the documentation for the encoders and is outside the scope of this document.

Building an MPEG video, especially for a large event, can take some time and should not be undertaken lightly as the effect on your host box of many CPU intensive encoders will not be good. However once a video has been created for an event it will be kept so subsequent viewing will not incur the generation overhead. Videos can also be included in notification emails, however care should be taken when using this option as for many frequent events the penalty in CPU and disk space can quickly mount up.

## **6.7. Options and Users**

The final area covered by the tutorial is the options and user section. If you are running in authenticated mode and don't have system privileges then you will not see this section at all and if you are running in un-authenticated mode then no user section will be displayed.

The various options you can specify are displayed in a tabbed dialog with each group of options displayed under a different heading. Each option is displayed with its name, a short description and the current value. You can also click on the '?' link following each description to get a fuller explanation about each option. This is the same as you would get from `zmconfig.pl`. A number of option groups have a master option near the top which enables or disables the whole group so you should be aware of the state of this before modifying options and expecting them to make any difference.

If you have changed the value of an option you should then 'save' it. A number of the option groups will then prompt you to let you know that the option(s) you have changed will require a system restart. This is not done automatically in case you will be changing many values in the same session, however once you have made all of your changes you should restart ZoneMinder as soon as possible. The reason for this is that web and some scripts will pick up the new changes immediately but some of the daemons will still be using the old values and this can lead to data inconsistency or loss.

One of the options you may notice in the 'System' tab allows you to specify the default language for your installation of ZoneMinder. Versions 1.17.0 and later support multiple languages but rely on users to assist in creating language files for specific languages. To specify a language you will have to give the applicable code, thus for UK English this is en\_gb, and for US English it would be en\_us, if no language is given then UK English is assumed. Most languages will be specified in this nn\_mm format and to check which languages are available look for files named zm\_lang\_\*.php in the ZoneMinder build directory where the parts represented by the '\*' would be what you would enter as a language. This is slightly unwieldy and will probably be improved in future to make it easier to determine language availability. On checking which languages are available it may be that your preferred language is not currently included and if this is the case please consider doing a translation and sending it back to it may be included in future releases. All the language elements are given in the zm\_lang\_en\_gb.php file along with a few notes to help you understand the format.

As mentioned above, you may also see a 'users' tab in the Options area. In this section you will see a list of the current users defined on the system. You can also add or delete users from here. It is recommended you do not delete the admin user unless you have created another fully privileged user to take over the same role. Each user is defined with a name and password (which is hidden) as well as an enabled setting which you can use to temporarily enable or disable users, for example a guest user for limited time access. As well as that there is a language setting that allows you to define user specific languages. Setting a language here that is different than the system language will mean that when that user logs in they will have the web interface presented in their own language rather than the system default, if it is available. Specifying a language here is done in the same way as for the system default language described above.

There are also four values that define the user permissions, these are 'stream', 'events', 'monitors' and 'system' Each can have values of 'none', 'view' or 'edit' apart from 'stream' which has no 'edit' setting. These values cover access to the following areas; 'stream' defines whether a user is allowed to view the 'live' video feeds coming from the cameras. You may wish to allow a user to view historical events only in which case this setting should be 'none'. The 'events' setting determines whether a user can view and modify or delete any retained historical events. The 'monitors' setting specifies whether a user can see the current monitor settings and change them. Finally the 'system' setting determines whether a user can view or modify the system settings as a whole, such as options and users or controlling the running of the system as a whole. As well as these settings there is also a monitor ids setting that can be used for non-'system' users to restrict them to only being able to access streams, events or monitors for the given monitors ids as a comma separated list with no spaces. If a user with 'monitors' edit privileges is limited to specific monitors here they will not be able to add or delete monitors but only change the details of those they have access to. If a user has 'system' privileges then the 'monitors ids' setting is ignored and has no effect.'

That's pretty much is it for the tour. You should experiment with the various setting to get the results you think are right for your. Naturally, letting thousands of events build up is not good for the database or your file system so you should endeavour to either prevent spurious events from being generated in the first place or ensure that you housekeep them strictly.



## 7. Camera Control

Version 1.21.0 of ZoneMinder introduces a new feature, allowing you to control cameras from the web interface and to some extent automatically. Pan/Tilt/Zoom (PTZ) cameras have a wide range of capabilities and use a large number of different protocols making any kind of generic control solution potentially very difficult. To address this ZoneMinder uses two key approaches to get around this problem.

1) Definition of Capabilities – For each camera model you use, an entry in the camera capabilities table must be created. These indicate what functions the camera supports and ensure that the interface presents only those capabilities that the camera supports. There are a very large number of capabilities that may be supported and it is very important that the entries in this table reflect the actual abilities of the camera. A small number of example capabilities are included in ZoneMinder, these can be used ‘as is’ or modified.

2) Control Scripts – ZoneMinder itself does not generally provide the ability to send commands to cameras or receive responses. What it does is mediate motion requests from the web interface into a standard set of commands which are passed to a script defined in the control capability. Example scripts are provided in ZoneMinder which support a number of serial or network protocols but it is likely that for many cameras new scripts will have to be created. These can be modelled on the example ones, or if control commands already exist from other applications, then the script can just act as a ‘glue’ layer between ZoneMinder and those commands.

It should be emphasised that the control and capability elements of ZoneMinder are not intended to be able to support every camera out of the box. Some degree of development is likely to be required for many cameras. This should often be a relatively straightforward task however if you have a camera that you want to be supported then please feel free to get in touch and I should be able to provide an estimate for how much effort this is likely to be. It is also the case that I have only been able to access this limited number of cameras to test against; some other cameras may use different motion paradigms that don’t fit into the control capability/script architecture that ZoneMinder uses. If you come across any cameras like this then please forward as much information to me as possible so that I may be able to extend the ZoneMinder model to encompass them.

### 7.1. Control Capabilities

If you have a camera that supports PTZ controls and wish to use it with ZoneMinder then the first thing you need to do is ensure that it has an accurate entry in the capabilities table. To do this you need to go to the Control tab of the Monitor configuration dialog and select ‘Edit’ where it is listed by the Control Type selection box. This will bring up a new window which lists, with a brief summary, the existing capabilities. To edit an existing capability to modify select the Id or Name of the capability in question, or click on the Add button to add a new control capability. Either of these approaches will create a new window, in familiar style, with tabs along the top and forms fields below. In the case of the capabilities table there are a large number of settings and tabs, the mean and use of these are briefly explained below.

#### ‘Main’ Tab

**Name** – This is the name of the control capability, it will usually make sense to name capabilities after the camera model or protocol being used.

**Type** – Whether the capability uses a local (usually serial) or network control protocol.

**Command** – This is the full path to a script or application that will map the standard set of ZoneMinder control commands to equivalent control protocol command. This may be one of the shipped example zmcontrol-\*.pl scripts or something else entirely.

**Can Wake** – This is the first of the actual capability definitions. Checking this box indicates that a protocol command exists to wake up the camera from a sleeping state.

**Can Sleep** – The camera can be put to sleep.

**Can Reset** – The camera can be reset to a previously defined state.

#### ‘Move’ Tab

**Can Move** – The camera is able move, i.e. pan or tilt.

**Can Move Diagonally** – The camera can move diagonally. Some devices can move only vertically or horizontally at a time.

**Can Move Mapped** – The camera is able internally map a point on an image to a precise degree of motion to centre that point in the image.

**Can Move Absolute** – The camera can move to an absolute location.

**Can Move Relative** – The camera can move to a relative location, e.g. 7 point left or up.

**Can Move Continuous** – The camera can move continuously in a defined direction until told to stop or the movement limits are reached, e.g. left.

#### **'Pan' Tab**

**Can Pan** – The camera can pan, or move horizontally.

**Min/Max Pan Range** – If the camera supports absolute motion this is the minimum and maximum pan co-ordinates that may be specified, e.g. -100 to 100.

**Min/Max Pan Step** – If the camera supports relative motion, this is the minimum and maximum amount of movement that can be specified.

**Has Pan Speed** – The camera supports specification of pan speeds.

**Min/Max Pan Speed** – The minimum and maximum pan speed supported.

**Has Turbo Pan** – The camera supports an additional turbo pan speed.

**Turbo Pan Speed** – The actual turbo pan speed.

#### **'Tilt' Tab**

Definition of Tilt capabilities, fields as for 'Pan' tab.

#### **'Zoom' Tab**

**Can Zoom** – The camera can zoom.

**Can Zoom Absolute** – The camera can zoom to an absolute position.

**Can Zoom Relative** – The camera can zoom to a relative position.

**Can Zoom Continuous** – The camera can zoom continuously in or out until told to stop or the zoom limits are reached.

**Min/Max Zoom Range** – If the camera supports absolute zoom this is the minimum and maximum zoom amounts that may be specified.

**Min/Max Zoom Step** – If the camera supports relative zoom, this is the minimum and maximum amount of zoom change that can be specified.

**Has Zoom Speed** – The camera supports specification of zoom speed.

**Min/Max Zoom Speed** – The minimum and maximum zoom speed supported.

#### **'Focus' Tab**

Definition of Focus capabilities, fields as for 'Zoom' tab, but with the following additional capability.

**Can Auto Focus** – The camera can focus automatically.

#### **'White' Tab**

Definition of White Balance capabilities, fields as for 'Focus' tab.

#### **'Iris' Tab**

Definition of Iris Control capabilities, fields as for 'Focus' tab.

#### **'Presets' Tab**

**Has Presets** – The camera supports preset positions.

**Num Presets** – How many presets the camera supports. If the camera supports a huge number of presets then it makes sense to specify a more reasonable number here, 20 or less is recommended.

**Has Home Preset** – The camera has a defined 'home' position, usually in the mid point of its range.

**Can Set Presets** – The camera supports setting preset locations via its control protocol.

## 7.2. Control Scripts

The second key element to controlling cameras with ZoneMinder is ensuring that an appropriate control script or application is present. A small number of sample scripts are included with ZoneMinder and can be used directly or as the basis for development. Control scripts are run atomically, that is to say that one requested action from the web interface results in one execution of the script and no state information is maintained. If your protocol requires state information to be preserved then you should ensure that your scripts do this as ZoneMinder has no concept of the state of the camera in control terms.

If you are writing a new control script then you need to ensure that it supports the parameters that ZoneMinder will pass to it. If you already have scripts or applications that control your cameras, the ZoneMinder control script will just act as glue to convert the parameters passed into a form that your existing application understands. If you are writing a script to support a new protocol then you will need to convert the parameters passed into the script to equivalent protocol commands. If you have carefully defined your control capabilities above then you should only expect commands that correspond to those capabilities.

The standard set of parameters passed to control scripts is defined below,

**--device=<device>** – This is the control device from the monitor definition. Absent if no device is specified.

**--address=<address>** – This is the control address from the monitor definition. This will usually be a hostname or ip address for network cameras or a simple numeric camera id for other cameras.

**--command=<command>** – This specifies the command that the script should execute. Valid commands are given below.

**--xcoord=<x>, --ycoord=<y>** – This specifies the x and/or y coordinates for commands which require them. These will normally be absolute or mapped commands.

**--width=<width>, --height=<height>** – This specifies the width and height of the current image, for mapped motion commands where the coordinates values passed must have a context.

**--speed=<speed>** – This specifies the speed that the command should use, if appropriate.

**--panspeed=<speed>, --tiltspeed=<speed>** – This indicates the specific pan and tilt speeds for diagonal movements which may allow a different motion rate for horizontal and vertical components.

**--step=<step>** – This specifies the amount of motion that the command should use, if appropriate. Normally used for relative commands only.

**--panstep=<step>, --tiltstep=<step>** – This indicates the specific pan and tilt steps for diagonal movements which may allow a different amount of motion for horizontal and vertical components.

**--preset=<preset>** – This specifies the particular preset that relevant commands should operate on.

The 'command' option listed above may take one of the following commands as a parameter.

**wake** – Wake the camera.

**sleep** – Send the camera to sleep.

**reset** – Reset the camera.

**move\_map** – Move mapped to a specified location on the image.

**move\_pseudo\_map** – As move\_map above. Pseudo-mapped motion can be used when mapped motion is not supported but relative motion is in which case mapped motion can be roughly approximated by careful calibration.

**move\_abs\_<direction>** – Move to a specified absolute location. The direction element gives a hint to the direction to go but can be omitted. If present it will be one of 'up', 'down', 'left', 'right', 'upleft', 'upright', 'downleft' or 'downright'.

**move\_rel\_<direction>** – Move a specified amount in the given direction.

**move\_con\_<direction>** – Move continuously in the given direction until told to stop.

**move\_stop** – Stop any motion which may be in progress.

**zoom\_abs\_<direction>** – Zoom to a specified absolute zoom position. The direction element gives a hint to the direction to go but can be omitted. If present it will be one of 'tele' or 'wide'.

**zoom\_rel\_<direction>** – Zoom a specified amount in the given direction.

**zoom\_con\_<direction>** – Zoom continuously in the given direction until told to stop.

**zoom\_stop** – Stop any zooming which may be in progress.

**focus\_auto** – Set focusing to be automatic.

**focus\_man** – Set focusing to be manual.

**focus\_abs\_<direction>** – Focus to a specified absolute focus position. The direction element gives a hint to the direction to go but can be omitted. If present it will be one of 'near' or 'far'.

**focus\_rel\_<direction>** – Focus a specified amount in the given direction.

**focus\_con\_<direction>** – Focus continuously in the given direction until told to stop.

**focus\_stop** – Stop any focusing which may be in progress.

**white\_<subcommand>** - As per the focus commands, except that direction may be 'in' or 'out'.

**iris\_<subcommand>** - As per the focus commands, except that direction may be 'open' or 'close'.

**preset\_set** – Set the given preset to the current location.

**preset\_goto** – Move to the given preset.

**preset\_home** – Move to the 'home' preset.

## 8. Mobile Devices

ZoneMinder has always had a minimal WML (Wireless Markup Language) capability to allow it to function on mobile phones and similar devices. However as of 1.20.0 this is now deprecated and has been replaced with a new XHTML – Mobile Profile mode as well as the default HTML4. XHTML-MP is a small, and limited, version of XHTML intended for mobile devices and is based on XHTML Basic. It does not contain scripting or other dynamic elements and essentially is a subset of HTML as most people know it.

The ZoneMinder XHTML-MP interface allows you to log into your installation via your phone or mobile devices and perform a limited number of tasks. These include viewing recent events, and monitoring live streams. However unlike the full interfaces these elements are presented as still images requiring manual refreshing. For now the XHTML-MP interface is presented as a prototype interface; rather than one offering full capabilities. As such, please feel free to make comments or offer suggestions via the forums on <http://www.zoneminder.com>. One final word on mobile mark-up standards in general and their application to ZoneMinder. Ideally I'd like to offer a WML2.0 interface. WML2.0 is a blending of WML1.3, which is traditional WAP, and XHTML. As such it offers the scripting that WML has traditionally included plus the better control of markup that is the realm of XHTML. Unfortunately so far I'm unaware of any devices that support WML2.0, certainly I've never had a phone that does. If you find out that a particular phone does support this then please let me know (or better still send me the phone!).

If you wish to use the XHTML-MP interface to ZoneMinder there is no extra configuration required to enable it per se. However ZoneMinder needs to be able to figure out what kind of content to deliver to particular browsers and so you have two choices. You can either edit zm.php and include a definition that corresponds to your phone describing a small number of basic capabilities, you will see a couple of examples already there, or you can use the comprehensive open source WURFL package available from <http://wurfl.sourceforge.net/>. You will need to download both the WURFL php files and the wurfl.xml file itself. WURFL is a resource containing information on the capabilities of a huge number of mobile phones, devices and browsers. Thus once it has matched your phone it can determine various capabilities it may possess. This means that ZoneMinder itself only has to deal with these capabilities and not the individual phone types.

To use WURFL you should install the php files in the same directory as ZoneMinder and then create a 'wurfl' sub-directory and ensure it is readable and writeable (or preferably owned by) your web server user. You should put the wurfl.xml file in there. One other thing you may need to change, as the xml file is quite large, is the 'memory\_limit' setting in php.ini as the default setting of 8Mb may be too small. Once you've done this you should find that your phone or device is recognised and if it can support XHTML-MP it will receive that interface. If your phone is very new, or you are using an old version of the XML file you might find that it is not present however. The WURFL library uses a caching strategy to avoid reloading the whole XML file each time so check if a sensible looking cache file has been created in the 'wurfl' sub-directory also check the wurfl.log in the same place.

The WURFL is a third party application and as such I am unable to offer support directly for it. If you feel your device is missing or incorrectly represented please contact the authors via their own channels. If on the other hand you have any comments on ZoneMinder on your device specifically please let me know and I would be pleased to hear about it.

As support for cookies in mobile devices is patchy at best, the groups feature is not fully implemented in the XHTML-MP views. Instead if there is a group called 'Mobile' already defined then that group will always be effective, if not then all monitors available to the logged in user will be visible,

## 9. Troubleshooting

Life eh? Nothing ever works first time does it? In case you are having problems here are some things to try. If these don't work then check the ZoneMinder FAQ at <http://www.zoneminder.com/faq.html> and then the forums at <http://www.zoneminder.com/forums.html> first and see if anyone has had the same problem in the past. If not then feel free to get in touch and I'll see if I can suggest something else. The best places to look for errors are in the system error log (normally /var/log/messages on RedHat), the ZoneMinder logs, and the web server log (/var/log/httpd/error\_log unless otherwise defined). There should be something in one of those that gives you some kind of tip off.

Some things to check.

- Device configuration. If you can't get your cameras to work in ZoneMinder, firstly make sure that you have the correct settings. Use xawtv or something like that to check for settings that work and then run `zmu -d <device_no> -q -v` to get the settings. If you can't get them to work with that then the likelihood is they won't work with ZoneMinder. Also check the system logs (usually /var/log/messages) for any video configuration errors. If you get some and you're sure they're not a problem then switch off ZM\_STRICT\_VIDEO\_CONFIG in `zmconfig.pl` or the 'options' tab.
- Start simple. Begin with a single monitor and single zone. You can run the `zmc` capture daemon from the command line as `'zmc --device 0'` (or whatever your video device is). If it returns immediately there's a problem so check the logs, if it stays up then your video configuration is probably ok. To get more information out of it use `debug` as specified below. Also check that the shared memory segment has been created by doing `'ipcs -m'`. Finally, beware of doing tests as root and then trying to run as another user as some files may not be accessible. If you're checking things as root make sure that you clean up afterwards!
- Web server. Ensure that your web server can serve PHP files. It's also possible that your `php.ini` file may have some settings which break ZoneMinder, I'm not a PHP guru but setting safe mode may prevent your PHP files from running certain programs. You may have to set configuration to allow this. Also since the daemons are started by your web server, if it dies or is shut down then the daemons may disappear. In this version the daemons are run under the control of a script which should trap expected signals but it is possible this doesn't cover all circumstances.
- One of the more common errors you can see in the log files is of the form 'Can't shmget: Invalid argument'. Generally speaking this is caused by an attempt to allocate an amount of shared memory greater than your system can handle. The size it requests is based on the following formula,  $\text{ring buffer size} \times \text{image width} \times \text{image height} \times 3$  (for 24 bit images) + a bit of overhead. So if for instance you were using 24bit 640x480 then this would come to about 92Mb if you are using the default buffer size of 100. If this is too large then you can either reduce the image or buffer sizes or increase the maximum amount of shared memory available. If you are using RedHat then you can get details on how to change these settings at [http://www.redhat.com/docs/manuals/database/RHDB-2.1-Manual/admin\\_user/kernel-resources.html](http://www.redhat.com/docs/manuals/database/RHDB-2.1-Manual/admin_user/kernel-resources.html)
- You should be able to use a similar procedure with other distributions to modify the shared memory pool without kernel recompilations though in some cases this may be necessary. Note, this error also sometimes occurs if you have an old shared memory segment lying around from a previous run that is too small. Use the `ipcs` and `ipcrm` system commands to check and remove it if necessary.
- If you get odd javascript errors and your web console or other screens come up with bits missing then it's possible that there is a problem with the PHP configuration. Since version 0.9.8 ZoneMinder has used short PHP open tags to output information, so instead of something like this `'<?php echo $value ?>'`, it will be something like this `'<?=$value ?>'` which is easier and quicker to write as well as being neater. More information about this directive can be seen at the following location, <http://www.php.net/manual/en/configuration.directives.php#ini.short-open-tag>. However although by default most PHP installations support this form, some will need to have it switched on explicitly. To do this you will first need to find your `php.ini` file (do a 'locate

php.ini' or 'find / -name php.ini'. Be aware however that sometimes you might find more than one, so ensure you identify the one that is actually being used. You will then need to find the line that starts 'short\_open\_tag = ' and change the Off value to On. This will correct the problem. However in some cases you may have explicitly switched it off, so that XML compliant documents can be more easily served, or you may even not have permission to edit the file. In this case you can go into the web directory of ZoneMinder and run 'sh retag.sh' which will replace all the short open tags in the files themselves with the longer variant. You will obviously have to remember to do this for each subsequent version of ZoneMinder that you install as well.

- Use debug. ZoneMinder has various debug in it that by default will go into your system log (via syslog). These will be of the form of

```
"Sep 14 14:50:11 localhost zma-0[1975]: INF [Front: 221000 - Processing at 4.26 fps ]"
```

where the zma-0 part identifies the daemon and the device it is running on. Entries with INF in are informational and not an error, if you see ERR then it is one, though not all are fatal. You can prevent this information from being emitted by setting the ZM\_DBG\_LEVEL\_zmc environment variable to -1 or less once things are working. If you want to run any of the daemons from the command line to test, setting ZM\_DBG\_PRINT to 1 will output the debug on the console. You can also use the USR1 and USR2 signals to increase or decrease the amount of debug being emitted.

- Paths. I admit it, the various paths in ZoneMinder can be bit of a nightmare. Make sure that they are all correct and that permissions are such that the various parts of ZoneMinder can actually run.
- Missing perl modules. There are various perl modules used by the various scripts. If you get errors about missing ones, the easiest way to install them is to type the following (you will probably need to be root),

```
perl -MCPAN -eshell
```

this will then (eventually, after some configuration if it's your first time) present you with a prompt. From there you can type install module, e.g. Archive::Zip and the rest should be more or less automatic as it will chase any dependencies for you. There may be some initial configuration questions it might ask you on startup if you've never run it before and to speed things up I would not install a new Bundle at this point (it can end up building you a whole new perl if you're not careful) if it asks you but everything else should be quite straightforward.

- Unsupported palettes. ZoneMinder currently is designed to use the simple palettes of greyscale and 24 bit as well as now the YUV420P and some other palettes. This should cover most cameras but it's possible that there are ones out there that might want to use more esoteric formats that ZoneMinder doesn't support. This will often show up as the capture daemon being unable to set picture attributes. If this occurs try using different palettes starting with greyscale and if you can't get anything to work let me know and I'll try and add it.
- USB bus problems. If you have multiple USB cameras on one bus then it can appear as if ZoneMinder is causing your cameras to fail. This is because the bandwidth available to cameras is limited by the fairly low USB speed. In order to use more than one USB camera with ZoneMinder (or any application) you will need to inform the driver that there are other cameras requiring bandwidth. This is usually done with a simple module option. Examples are `usb_alt=<n>` for the OV511 driver and `cams=<n>` for CPIA etc. Check your driver documentation for more details. Be aware however that sharing cameras in this way on one bus will also limit the capture rate due to the reduced bandwidth.
- Incorrect libjpeg.a detection. It seems to be the case that in some cases the library file libjpeg.a is reported as missing even when apparently present. This appears to actually be down to the g++ compiler not being installed on the host system. Since ZoneMinder contains both C++ and C files you need to be able to compile both of these file types and so usually need to ensure you have gcc and g++ installed (though they are often the same binary).

- Httpd and zms memory leaks. It has been reported by some users with RedHat 9 that the zms process fails to terminate correctly when the controlled window is killed and also that it, and it's associated httpd process, continue to grow in memory size until they kill the system. This appears to be a bug in early versions of apache 2. On other systems it may appear that zms is leaking and growing. However what grows is the total and shared memory size while the non-shared memory size stays constant. It's a little odd but I think what is happening is that as zms picks images out of the shared memory ring buffer to display, as each slot is read the size of that bit of memory is added to the shared memory total for the process. As streamed images are not read consecutively it's a semi-random process so initially most of the buffer slots are new and the shared memory size grows then as time goes on the remaining unaccessed slots reduce until once all have been read the shared memory use caps out at the same size as the actual segment. This is what I would have expected it to be in the first place, but it seems to do it incrementally. Then once this total is hit it grows no further. As it's shared memory anyway and already in use this apparent leak is not consuming any more memory than when it started.
- Cambozola. There appears to be an issue with recent versions of Cambozola that causes image corruption in the stream. If you are getting this then I suggest you stick with version 0.22 which is available from the Downloads section of [www.zoneminder.com](http://www.zoneminder.com).

Also, if you are using IE under Windows and get lots of annoying clicks when various windows refresh then you'll need to edit your registry and remove the value for HKEY\_CURRENT\_USER\AppData\Local\Microsoft\Internet Explorer\Navigation\NoIEClick.reg or download the registry script to do it for you from <http://www.zoneminder.com/downloads/noIEClick.reg>



## **10. Change Log**

### **10.1. Release 1.21.0**

Addition of camera control, plus several bugfixes.

- FEATURE – Added support for Pan/Tilt/Zoom and general camera control.
- FIX – The montage view layout has been modified to allow better dynamic layout of windows. Views should now be laid out in a more logical arrangement. This is a relatively temporary change and the montage view will shortly be rewritten to use flowing 'div' tags which should add more flexibility and be less complex.
- FIX – All stream views now have an 'alt' tag to highlight which monitor they should be displaying.
- FIX – Detection of which markup language to use, HTML or XHTML-MP has now been optimised to ensure that the test only happens once per session.
- FIX – Some constants were defined unquoted, this has now been corrected.
- FIX – The zmtrigger.pl script had an old, and incorrect, initial section using constants that were no longer valid. This has now been fixed.
- FIX – The regular expression patterns used to parse the zm.conf file have been modified to ensure that they should always work.
- FIX – In previous versions it was possible for a process to die and not be reaped by zmdc.pl. This could have resulted in processes remaining as zombies resulting in them not being restarted after crashing. This has now been fixed so all dying processes will be caught and handled.
- FIX – The frame view has been restructured to ensure that it has a consistent look and does not display unwanted wrapping.
- FIX – A couple of remnant hard coded text elements have been replaced with tokens as they should have been originally.
- FIX – Previously separate 'object' and 'embed' tags were used for Internet Explorer and non-IE browsers. These have been merged so that browsers will use whichever tag is appropriate. Any player controls that were present should now be hidden as well.
- FIX – A problem was present whereby the Maximum FPS set in the bandwidth settings was not being respected in the live streams. This is now fixed.
- FIX – If users were created with restricted monitor ids, it was sometimes possible that permission errors would still be issued if they tried to view streams or other elements associated with monitors in their list. This has now been corrected such that any restrictions are applied correctly.
- FIX – Users created with only 'view streams' permissions were presented with a 'permission denied' error in the area of the Watch window normally containing the recent events list. Whilst this was technically correct it was unnecessary and untidy, and has now been changed just to be blank.

### **10.2. Release 1.20.1**

Mostly bug fixes, large and small with a couple of minor features included.

- FIX – A dependency on the regular expression library was introduced in 1.20.0 which caused some people to have build problems. This library has traditionally been necessary to support network cameras but not otherwise. This situation has now been restored.

- FEATURE – Added ZM\_RAND\_STREAM option. This option adds a time code onto the url of each stream to prevent it from being cached which had caused some broken image problems with some browsers, notably Mozilla.
- FIX – Made zms check ZM\_OPT\_AUTH before loading user details. This should have been in there in 1.20.0 but was omitted and should fix the issue where streams did not work with authentication off.
- FIX – There was some debug code left behind in zm\_xhtml.php. This was unnecessary and has been removed.
- FIX – Fixed user sql, added debug and wrapped in check for libcrypto in zm\_user.cpp. This should correct bogus loading of user data which may have affected some people. You can also now just bump up the debug level to see what the auth strings being used are.
- FIX – The XHTML console page now uses the mobile group as it should have in 1.20.0
- FIX – Modified database username to be binary. You need to run the zmalter-1.20.0 sql script as usual to change your Users table to disallow case-insensitive checking which may have been breaking some people's streams.
- FIX – Fixed incorrect constant definitions in zmtrigger.pl. This script had not been updated along with the other scripts.
- FIX – Fixed bogus double .jpg suffix on diagnostic images, also included them (if they exist) in frame view.
- FIX – Corrected broken check for libcrypto (the check happened before any definition) causing build problems for some people who do not have MD5 library installed.
- FIX – Added permissions mode to mkdir in zm\_actions.php to remove php warning.
- FIX – Added space before -m in zmu command in zm\_actions.php
- FIX – Added quotes around brightness etc SQL in zm\_actions.php to avoid errors when values are empty.
- FIX – Added line length to fgets in zm\_config.php.z to prevent php warning
- FIX – Slightly enlarged a couple of window sizes in zm\_config.php.z to work better with different browsers.
- FIX – Defined empty array in html\_view\_states to prevent php warnings.
- FEATURE – Console window now sizes itself according to how many monitors in list, though there is a minimum size.
- FIX – Corrected bug in zmfilter.pl.z which meant that images were not always correctly uploaded.

### **10.3. Release 1.20.0**

Improved and added features, several minor bug fixes.

- FEATURE – Certain configuration (Mostly database settings) is now stored in a new file zm.conf. This means that database access settings can be changed without recompilation. It also allows the creation of ZoneMinder rpms. Watch zoneminder.com for details. Thanks for forum user 'oskin' for his work on this.
- FEATURE – The WML interface is now deprecated and the XHTML-MP interface is the new supported interface for mobile devices.
- FEATURE – Monitor groups have now been added allowing subsets of monitors to be viewed independently.

- FEATURE – A generic external triggering interface has been included via the zmtrigger.pl script. A new monitor function 'Nodect' has been added to support this.
- FEATURE – Interaction between the web pages and the streaming daemons and other utilities has previously been not as secure as it could have been and open to possible abuse. This has now been addressed and zms and zmu both now use (optional) authentication strings to validate access. You need to have openssl installed so that the MD5 libraries can be linked. See the ZM\_AUTH\_METHOD and ZM\_AUTH\_SECRET configuration items for further details.
- FEATURE – The maximum daemon restart delay in zmdc.pl was previously fixed at 15 minutes. This may have been too long for some users, for example if power has failed to a camera then a 15 minute delay on restoration is not desirable. This maximum is now configurable via the ZM\_MAX\_RESTART\_DELAY configuration item.
- FEATURE – The web files have been changed to use the newer style autoglobals, e.g. \$\_SERVER rather than \$HTTP\_SERVER\_VARS. This should enable use on PHP5 without any modification.
- FIX – The use of two database users has been somewhat redundant for a number of versions now. In 1.20.0 there is only one database user. The zmupdate.pl script unfortunately cannot handle the migration as it needs to access the database so you should make a note of the username and password of the privileged user and then re-enter that using zmconfig.pl when rebuilding ZM.
- FIX – The zmupdate.pl script previously held a database connection open for days at a time but only used to use it periodically. This has now been changed to be only open while in use.
- FIX – Debug output and it's relationship with environment variables etc was previously broken. This has been tidied up and made much easier to use and understand.
- FIX – A number of SQL queries have been analysed and optimised to run much faster.
- FIX – The monitor status was not always being reported correctly in the monitor watch window. This has been corrected.
- FIX – Image numbering in the zmf daemon was sometimes wrong if more or less than three significant digits were used. This has been corrected.
- FIX – Image capture timeouts used by zmwatch.pl to restart apparently frozen zmc processes were being calculated incorrectly on occasion. This was causing some unnecessary processes to be restarted. This calculation has been fixed.
- FIX – Complete DOCTYPE headers were added to HTML output and some HTML was tidied up to be more compliant.
- FIX – There was a problem with the interaction between monitor statuses and the status web window. This meant that sometimes the window did not pop to the front, or play the alarm sound, properly. This has been corrected.
- FIX – Some network cameras send data in a format which was previously not recognised by the regular expression engine. This has been modified to allow these cameras (NC1000 etc) to function with ZoneMinder.
- FIX – A bug in event streaming when events are of very short duration has been fixed. Thanks to forum user 'reza' for spotting this one.
- FIX – A possible exploit in the login page was identified and has now been fixed. Thanks again to forum user 'reza' highlighting this problem also.

#### **10.4. Release 1.19.5**

Various miscellaneous fixes and features.

- FIX – Sorting event lists by duration was broken and has now been corrected.

- FEATURE – The zmfux utility previous corrected file permissions on video device files only. This has been modified to do likewise to the X10 device serial port if enabled.
- FIX – The modification suggested by forum user 'oskin' has been incorporated into the code to try and reduce or remove video for linux errors.
- FIX – The remote network camera parsing code has been patched to try
- FIX – The error reported when a 'shmget' call fails has been changed to include further information about the cause.
- LANGUAGE – Fixed missing semicolon in German language file.
- FEATURE – Added '<<' and '>>' links to the page selector in the events list as suggested by forum user 'unclerichy'.
- FEATURE – Brightness, colour, hue and contrast are now saved persistently for a monitor rather than being reset each time the system is restarted. This feature is based on a patch submitted by forum user 'oskin'.
- FEATURE – In previous versions the events folder has been keyed by the monitor name. This has caused problems in the past with various characters appearing which are legal in names but not in filesystems. From this version all files related to monitors are keyed on the monitor id rather than the name. To help you navigate through these files the monitor name still exists but as a link only. Please ensure you run zupdate.pl to update your events directory.
- FEATURE – You may now optionally have thumbnail images in your event lists. To enable this functionality set ZM\_WEB\_LIST\_THUMBS on in Options->Web. You can also control the width or height of these thumbnails but should only set one dimension only and leave the other blank or zero.
- FEATURE – You can now specify how many image thumbnails appear across and down the page in the event stills listing. In Options->Web set the ZM\_WEB\_FRAMES\_PER\_LINE and/or ZM\_WEB\_FRAME\_LINES options.
- FEATURE – ZoneMinder uses ffmpeg (<http://ffmpeg.sourceforge.net/>) for video generation and processing. Recently a new version (0.4.9-pre1) was released which changed the interface that ZoneMinder uses and so broke compilation. This version will detect which version of ffmpeg you have and compile accordingly.
- FEATURE – You can now specify a prefix for events generated by particular monitors. This will replace the default 'Event-' one.
- FEATURE – If you use filters to send event notification emails you can now have them sent in HTML format. This is done automatically if your mail body includes a '<html>' token,
- FEATURE – An experimental feature has been added which lets you view several events in sequence. In event listing you can check the events you want to view and then click the 'View' button. This will allow you to navigate through only those events in the normal manner (via Prev and Next links) but also to view them in sequence by clicking on the 'Play All' link. This will replay each event and then automatically move onto the next one. You can stop this progression at any time by pressing 'Stop' (which only stops the sequence and not the currently playing event). The timing of the replay is done depending on the calculated length of the event (plus one second) and so may not exactly correspond to the real event length. In particular this is unlikely to work if replaying events using MPEG video and buffering players as the timing will likely be incorrect. If you are viewing an event but haven't checked any in the list the 'Play All' button will just work down the current event list.
- FIX – A default php error level excluding notice warnings is now explicitly set.
- FEATURE – Previously events have been created even if only one frame has generated an alarm. This has not always been desirable as sometimes glitches and flickers create large numbers of events, however no mechanism existed for limiting this. In this version you can now specify the minimum number of consecutive alarmed frames that are necessary to

create an event. This is the 'Alarm Frame Count' described above. Note that if an alarm is in progress single isolated alarmed frames will still prolong it and the count only applies to the initial frames that would cause the event.

## **10.5. Release 1.19.4**

Language fixes and updates.

- FIX – The US English language file was recursively including itself rather than the UK English file as the base language.
- LANGUAGE – The Brazilian Portuguese language file detailed in the previous release has actually been included in this one.
- LANGUAGE – The Argentinian Spanish, Polish and Italian translations have all been updated with tokens introduced in version 1.19.3.

## **10.6. Release 1.19.3**

Minor tweaks, fixes and language updates.

- FEATURE – All stills views now use the single image mode of zms rather than spawning off a zmu process to write an image which is then read. This reduces complexity of double buffering significantly and also reduces the chance of errors caused by multiple simultaneous image generation.
- FEATURE – The generated MIME types when creating streamed video were previously assigned by zms depending on which of a limited number of output formats was specified. This has now been changed so that the ffmpeg libavformat library itself now generates these identifiers. The consequence of this is that many more video formats supported by your version of ffmpeg should now be available via zms.
- FEATURE – When viewing a single frame of an event you can select a 'stats' link to view the statistics that apply to that frame, if you have the RECORD\_EVENT\_STATS option switched on. This can be used to help configure your zones for optimal motion detection. Previously only pixel count values were displayed here which made it difficult to configure zones configured in percentage terms. These values are now displayed in both pixel and percentage terms to assist in zone configuration. Note that the percentage values are based on the current size of the zone so if this is changed then the value displayed will not be applicable at the time of event generation.
- FIX – When doing motion detection an extra blob, that could never be removed, was sometimes included. This could have caused false triggering and has not been corrected.
- FIX – A problem was reported whereby when using bulk frame records to reduce database load the last frame record was not written. Replaying the event via the web interface resulting in the event being truncated. A correction has been made so prevent this and ensure that the last frame of an event is always recorded.
- FIX – If an analysis daemon terminates abnormally or the host computer crashes then events can be left in a state whereby they effectively have zero length and are useless. A change to zmaudit.pl was made such that any 'open' events such as this which have not been updated for at least five minutes are closed and updated to reflect their actual content so that they may be viewed or saved. Events recovered in this way are named with a '(r)' mark to help identify them.
- FIX – In more recent versions of MySQL the password hash generated is 50 characters long, which overflows the previous password field in the database which was only 32 characters long. This field has been extended to 64 characters to accommodate this.
- FIX – The montage view had an error whereby the refresh timeout for stills was mislabelled causing continuous refresh attempts which rendered the view mostly unusable. The constants in question are now correctly referenced.

- FIX – The default, bandwidth specific, rate and scales were not always used as the records in the database were misnamed. This is now corrected though you may need to reset the values that were used previously as these will be lost if they had been changed.
- FIX – It was previously the case that old images could be left in the 'images' directory for a long period, sometimes resulting in incorrectly assuming correct operation. A fix was made to zmaudit.pl which modified the previous clean up of old WAP images so that any old images left in this directory are removed after a short period. Please ensure that if you have customised the web interface and have images you wish to keep that they are not left in the temporary images folder as they will now be deleted.
- FIX – A JavaScript error in the Zone configuration screen was identified and fixed.
- LANGUAGE – A Brazilian Portuguese translation has been supplied by Victor Diago and is available by selecting 'pt\_br' as the language type.
- LANGUAGE – Updated versions of the Dutch and Argentinian Spanish translations have been included.

## 10.7. Release 1.19.2

Minor features, fixes and language updates.

- FEATURE – The default replay rate and live and event scale settings are now configurable on a per bandwidth basis rather than globally. This allows you to view at full scale when you have high bandwidth and at smaller scales when you do not have so much resource. You will need to re-configure your previous defaults as they will be lost.
- FEATURE – Filters can now include a specification of the preferred sort order of the results.
- FEATURE – Filters can now include a specification to limit the results to a predefined maximum
- FEATURE – Two new filter elements have been added. These are disk blocks and disk percentage. These are event independent and return the amount of disk space used on the event partition in terms of disk blocks or percentage as returned by *df(1)*. Thus filters using these criteria will either match all events or none at all depending on the disk usage at the time of filter execution. The addition of these terms along with the ability to sort and limit filter results now means it is possible to create a filter that will automatically clear out old events once disk usage exceeds a certain value. Included in the database schemas for both new installations and upgrades is a sample filter called PurgeWhenFull which can be used to do this. It is initially not set to do anything automatically so if you want to use it, you should load it into the filter selection window, modify it to your taste and then save it, selecting 'auto delete'. Please note that filters created using disk related terms to delete events should always contain a limit term also otherwise it is possible for all events to match and thus be deleted. Using a limit ensures that only a small number are affected at any one time.
- FEATURE – Filters can now be defined to automatically execute an external script or program of your choosing. This can be specified when the filter is saved. Note that for security reasons this cannot be just any arbitrary command but must be readable and executable by your web server effective user. The script or program you specify here will be executed in the events root directory once for each event and will be passed one parameter containing the relative path to the event directory. This will normally be of the form *<MonitorName>/<EventId>* so it is possible to determine both the monitor and event in question from the path. Note also that a flag is set per event as with other auto actions indicating that an executable script has been run on that event and so to exclude it from subsequent matches. However if you have several filters all with executable scripts you will find that only the first gets executed as the flag will be set following successful completion and so no further scripts will be run on that event. Successful completion is indicated by the script returning a zero exit status, any other status indicates an error and the *executed* flag will not be set.
- FIX – In some circumstances temporary diagnostic images were being saved instead of highlighted analysis images. This is now corrected.

- FIX – When viewing a list of frames in an event, the link to the diagnostic image was incorrect. This is now fixed.
- FIX – The *Archive* link from the monitor watch window has been fixed. Previously this generated a bogus window.
- FIX – The zone definition have been updated so that selecting the various types of zones etc only disables those options you no longer have access to rather than wiping them out entirely. This is also true of the zone when saved. Thus you can now more easily change a zone to be temporarily inactive for example and have your previous active settings restored in the future.
- FIX – Selecting an event from the list generated by a filter that included a Monitor Name term did not previously work properly. This is now fixed.
- FIX – A number of the constants used internally have been renamed to be more consistent. Hopefully nothing is broken!
- FIX – Following notification of a potential vulnerability in zms by Mark Cox, all non-trivial string and buffer copies are now limited by the maximum size of the destination. Mark has also asked me to include the following notice relating to this, which I am very happy to do.
 

*"This issue was discovered by Mark J Cox <[mark@awe.com](mailto:mark@awe.com)>. The Common Vulnerabilities and Exposures project ([cve.mitre.org](http://cve.mitre.org)) has assigned the name CAN-2004-0227 to this issue."*
- LANGUAGE – An additional Italian language translation has been added. One, by Davide Morelli, was included in 1.19.1 but not announced. However like buses another one has come along, from Tolmino Muccitelli, and so they are both now present. The original translation is accessible by selecting *it\_it* as the language whereas the new one is *it\_it2*. I would prefer if they were merged as two versions of one language is not easy to maintain when I don't know what the differences mean!
- LANGUAGE – A version of Argentinian Spanish by Fernando Diaz has also been included and is accessible by setting your language to *es\_ar*. As with all the language translations I cannot vouch for the completeness or accuracy of the language files so feel free to feedback any updates you think should be made.
- NOTE – None of the non-English language files in this release do not contain any translations of the new, or modified, tokens which have been introduced in this release. All new or modified tokens are included in the language files in English. There will shortly be a point release which includes these language updates assuming I can get translations of them in a reasonable timescale.

## 10.8. Release 1.19.1

Minor bugfixes and enhancements.

- Ffmpeg Configure Changes. The configure script has been modified to look for the ffmpeg libraries in their installed location rather than in a build directory. This is to avoid having to build the library when it might already be installed.
- Pcre Configure Changes. The configure script has been modified to look for the pcre.h header file in both `/usr/include` and `/usr/include/pcre` rather than just the latter as previously.
- Remote Image Parsing. Further improvements have been made to handle additional patterns of images with differing styles of terminations or none at all.
- Event Image Numbering. An additional configuration option (`ZM_EVENT_IMAGE_DIGITS`) has been added to allow the user to define how many significant figures should be used to number individual event images.
- Frame Listing Timestamp Bug. Fixed a bug where in the event frame listing view the timestamps were not correctly displayed.

- Event Filters Bug. Fixed (again) a bug where several fields used in event filters did not generate valid database queries.
- Zmu Device Authentication. Removed the previous requirement to pass in a username and password to zmu when just querying a device as this was slightly broken and was unnecessary anyway.

## 10.9. Release 1.19.0

Some major enhancements and bugfixes.

- MPEG video streaming. ZoneMinder now supports true video streaming if configured with the *-with-ffmpeg* option. This allows one or both of live or event streaming to be in this format rather than motion JPEG style as before. Note however that is still somewhat experimental and may not work on your system. The reason for this is due to the variation in plugins and video movie formats. Currently I have got it working well with browsers on Windows platforms using the Windows Media Player plugin and the 'asf' video format. I have also managed to get event streaming working on Mozilla using mplayer (I think) though it jumps in and out of it's place in the window a bit. I would appreciate any feedback or advice on formats and plugins that work on your system. Also note that video streaming tends to get buffered before being displayed. This can result in the 'live' view being several seconds delayed.
- Motion JPEG Capture. Previously image capture from network devices has been limited to single stills capture only. This has now changed and if you entered a remote camera path that returns the *multipart/x-mixed-replace* MIME type then this will be parsed and images extracted from the stream. This is much faster than before and frame rates can be as fast now with network cameras as with capture cards and video. This feature also has the side-effect that one ZoneMinder installation can use another as a remote video source.
- NPH Streaming. After months of frustration I have finally figured out why streams were corrupted using Cambozola versions after 0.22. It turned out that apache was injecting characters into the streams which was screwing up the headers. I believe this to be because the initial header had no content-length header, as the length is indeterminate. So I have added a zero content length header which I believe fixes the problem though perhaps not in the best way. I have also made the installation link the existing zms binary to nph-zms so that you can now use zms in non-parsed-header mode. If it detects it is in this mode then the content-length header is not output, though several other additional ones are. In nph mode the false character injection seems to disappear so I suspect this is a better way to use zms.
- Bulk Frame Records. With the recent advent of the 'Record' and 'Mocord' modes a lot of people have started using ZoneMinder as a pseudo-DVR. This meant that a lot of database activity was taking place as each captured frame required its own entry in the database. The frames table has now been reorganised so that 'bulk' frames may be written at defined intervals to reduce this database activity. The records act as markers and individual frame timings are interpolated in between. Bulk frames are only used when no alarm or motion detection activity is taking place and normal frame records are kept otherwise.
- Event List Ordering and Scrolling. It was previously the case that the 'Next' and 'Prev' buttons on the event view did not always go to the event that was expected and sometimes disappeared altogether. This behaviour has now been modified and these buttons will now take you to the next and previous events in the list which the event was selected from. Thus if the list was sorted on ascending scores then the 'next' event is the one below which has a higher score etc. A possibly counterintuitive side effect of this is that as the default list is sorted by descending time the 'next' event is the one below in the list which will actually be earlier and the 'previous' event is later. So long as you remember that next and prev refer to the order of the list you should be ok.
- Zone Percentage Sizes. Zone motion detection parameters can be defined either in terms of total pixels or as a percentage. This percentage was defined relative to the size of the image as a whole. However this was difficult to calculate or estimate especially with several zones of varying sizes. In version 1.19.0 this has been changed so that the percentage relates to the size of the zone itself instead. This should make calculations somewhat easier. To



convert your existing zones you can run `zmupdate.pl` with the `-z` option, though this should be done only once and you should backup your database beforehand in case of error.

- Console View System Display. The console display was slight revamped to indicate disk space usage (via the `'df'` command) on the events partition,
- Zone Form Validation. Changes applied in version 1.18.0 to prevent invalidate entries in the zone definition form actually had the opposite effect due to JavaScript treating everything as a string and not a number (e.g. 5 is greater than 123). This is now corrected.
- Default Rate and Scales. You can now specify (in the options dialog) the default scale you would like to view live and event feeds at. You can also give a default rate for viewing event replays.
- More Rates. Additional faster rates have been included, up to 100 times.
- Frame Buffer Size. Previously it was possible for frames being sent from the analysis daemon to the frame server to exceed the defined maximum buffer size in which case the write would fail. It is now possible to define a larger size if necessary to prevent this. Note that you may have to adjust your system configuration to accommodate this. For further details check the help for the `ZM_FRAME_SOCKET_SIZE` option.
- Filter Name Duplication. Following recent changes to the filters table, several people reported that when saving filters they actually got a duplicate. This resulted in several copies of filters all with the same name as the constraint on unique filter names was not present. Well it is now so when upgrading your database all the filters will be renamed from `'myfilter'` to `'myfilter_<id>'` where `'<id>'` is the id number in the database (which is then removed). In general the higher the id number the more recent the filter. So you should go through your filter list deleting old copies and then rename the last one back to it's original name.
- Filter Form. Problem were reported with the filtering form where several selections generated SQL errors. This is now fixed.
- Filter Image Attachments. A fix was made to `zmfilter.pl` to prevent it trying to attaché alarm images to non-alarm events.
- Video Rate Specification. A fix was made to `zmvideo.pl` that corrected a problem with no default frame being used if none was passed in.
- RBG->BGR Black Screen. Fixed an issue with black screens being reported in RGB24 mode if RGB->BGR invert was not selected.
- Monitor Deletion. Fixed a problem with event files not being deleted when monitor was.
- A translation for the Dutch (nl\_nl) language has been included.

## **10.10. Release 1.18.1**

Minor bugfixes.

- Filter Monitor Name Bug. A bug was present in the previous release where monitor names where not correctly handled in filters. This is now fixed.
- Database Upgrade Change. Users upgrading from releases prior to 1.18.0 please note that now as part of the upgrade process all your filters will have any automatic actions unset. This is because the previous affinity to a particular monitor has now been removed and you may be left with several filters all doing the same thing to all of the events or have filters which for instance delete events on only one monitor but which now would delete them for all of them. It is recommended that you review your list of saved filters and delete duplicates before adding any monitor specific terms and resetting the actions for any that remain.

## **10.11. Release 1.18.0**

Major optimisations, important new features and some bugfixes.

- Optimisations and Performance Improvements. This release contains several major performance improvements in various areas. The first of these is that image processing for YUV style input formats are now pretty much handled at almost the same speed as native RGB formats. As this is what the capture daemons spend most of their time doing, the improvement helps reduce the amount of CPU time by a significant degree. Application of these changes also highlighted a bug that had existed previously in YUV conversion which caused incorrect conversions for certain values. The other two main areas of optimisation are in the Blend and Delta image functions. Normally when doing motion detection the analysis daemons spend about 99% of their time comparing a captured image with the reference image and then blending the two ready for the next capture. Both of these functions have been significantly improved. In previous versions there were two options for calculating image deltas (or differences), a simple RGB average and a Y channel calculation. Historically the RGB one was faster however with the optimisations the Y channel calculation (which is more accurate) is now 15-20% faster and so has become the default though you can select either method by the ZM\_Y\_IMAGE\_DELTAS configuration option. A new method of image blending has also been added which is up to 6 times faster than the old one which is retained for compatibility and because in some unusual circumstances it may still be more accurate (see the ZM\_FAST\_IMAGE\_BLENDS option for details). Altogether these optimisations (along with other common sense ones such as not maintaining a reference image in 'Record' mode where it is not used) significantly reduce the CPU load for most systems, especially when alarms are not in progress. If an alarm is detected then a lot of file system and database activity takes place which is limited by the speed of these resources so the gain will not be as much.
- Remote Authentication. This document has previously indicated that basic authentication for network cameras could be used by entering a hostname of the form of <user>:<pass>@<hostname>. This was not actually the case as the relevant authentication header was never sent. This is now fixed and addresses of this form can now be used.
- Filter Date Parsing. The zmfilter.pl date parsing now correctly reports when dates or times which it cannot parse are used.
- Monitor Independent Filters. Previously filters were closely tied to a monitor and a new filter had to be created for each monitor. This has now changed and filters can now specify an associated monitor in the same way as other parameters. Links have now been added to the main console view to allow you to view lists of events from all monitors in one and saved filters can now affect as many or as few monitors as you wish. **IMPORTANT:** Please note that as part of the upgrade process all your filters will have any automatic actions unset. This is because the previous affinity to a particular monitor has now been removed and you may be left with several filters all doing the same thing to all of the events or have filters which for instance delete events on only one monitor but which now would delete them for all of them. It is recommended that you review your list of saved filters and delete duplicates before adding any monitor specific terms and resetting the actions for any that remain.
- New Filter Operators. Two new filter operators and their inverse have been added. You can now indicate whether a value is in a set of other values, for example 'cat' is in the set of 'cat, dog, cow, horse'. You can also use regular expressions so 'cat' matches '^c.\*'. The 'not in set' and 'not matches' operators are also available.
- Additional Scales. Enhancements to the scaling algorithm mean that non binary scales are now just as easy to apply, thus new scales such as 0.75x have been added. Others can be easily included if necessary.
- Montage Sizing. The montage view allows you to view all of your active cameras in one window. However if your cameras are different sizes then this becomes very untidy. You can now constrain the image size of each monitor in this view to a fixed size with the ZM\_WEB\_MONTAGE\_WIDTH and ZM\_WEB\_MONTAGE\_HEIGHT configuration options. Monitor images will be enlarged or reduced as necessary.
- Compact Montage. The traditional montage view includes individual small menus for each monitor and a status display. This results in a somewhat cluttered display and the refreshing of the status displays may generate more accesses than desirable. Using the

ZM\_WEB\_COMPACT\_MONTAGE configuration option allows this montage view to only include the monitor streams and one overall menu bar with no status displays.

- Monitor Name Constraint. The name given to a monitor is used in file paths and several other areas. Thus it is important that it follows certain conventions but up until this release these names were unrestricted. The monitor form now limits monitor names to alphanumeric characters plus hyphen and underscore.
- Timestamp Change. Traditionally ZoneMinder has time-stamped each image as it is captured. This ensures that all images have their capture time recorded immediately. However there are several side-effects which may be undesirable. Firstly the time and resource is spent time-stamping images that are not recorded and which are discarded, secondly the timestamp is included in any motion detection and may potentially trigger an alarm if detection parameters are very sensitive. The third effect is that as the timestamp is added to the image at it's native resolution, if the image is scaled then the timestamp is scaled also. This may not be a problem for enlargement but if the image size is reduced then it may become illegible. This version now allows you, via the ZM\_TIMESTAMP\_ON\_CAPTURE configuration option, to indicate whether the timestamps should be added on capture, as before, or only added when the image is viewed or recorded. Setting it to this later value allows timestamps to be added to scaled images. This is little performance impact either way.
- Scaleable Stills View. The stills view of a monitor (when streaming is not available or desired) is now scaleable in the same way as the streamed view.
- Double Buffered Stills View. The stills view has now been restructured to allow a double buffering approach. Thus a new image is loaded in the background and only written to screen when complete. This removes the refresh flicker that means that the screen blanks periodically however uses more JavaScript so may not be suitable for all platforms. Whether ZoneMinder uses double buffering or not is controlled by the ZM\_WEB\_DOUBLE\_BUFFER configuration option.
- Fixed Length Event Bug. A bug was reported whereby the fixed length events that could be specified for use in Record or Mocord mode could sometimes result in events a multiple of that length. So events that were meant to be 15 minutes long could sometimes be 30 or even 45 minutes. This was especially the case with monitors that had low frame rates. This is now fixed.

## 10.12. Release 1.17.2

Minor features, bug fixes and additional languages.

- Pending Process Bug. A bug was found whereby a process that was scheduled to be started in the future (due to repeated failures) would drop out of the pending queue if a further explicit restart was attempted. This is now fixed.
- Strsignal Function. The strsignal function was included from version 1.17.1 however this is not ubiquitous on all distributions. The existence of this function is now tested for by the configure script and it is not used if not present.
- Add Max Alarm Threshold. Previously the alarm threshold (which is the amount a pixel has to differ from it's counterpart in the reference image) existed only in a 'minimum' form meaning pixels that were more different matched. A maximum has now been added to assist in screening out large changes in brightness. In addition to this a number of new consistency checks have been added to the zone definition form to try and prevent bogus or invalid settings.
- Diagnostic Zone Images. A regularly requested feature is that of adding extra information to allow diagnostics of the process of image detection. This has previously been somewhat hit and miss but in this version a new configuration option ZM\_RECORD\_DIAG\_IMAGES has been included to allow this. This option will generate several images for each captured frame in an alarm including each reference image and a series of images containing the image differences at various stages in the process. It is not possible to record these for the image

prior to an alarm but those following it are included and should assist in tuning the zones to provide optimal motion detection.

- Event Images Renamed. The capture and analysis images recorded during an event have been renamed from capture-???.jpg to ???-capture, and from analyse-???.jpg to ???-analyse.jpg. This is to allow all images (including diagnostic ones) to be associated with the frame sequence number more easily. This means that old events will no longer be able to be viewed as the wrong image will be being searched for. To avoid this you can use the new 'zmupdate.pl' utility to rename all your old images by doing 'perl zmupdate.pl -r' as an appropriately privileged or root user.
- Version checking. ZoneMinder will now optionally check for new versions of itself at zominder.com. This is done with a simple http get and no personal information otherwise than your current version of ZoneMinder is transmitted or recorded. If new versions are found you may be alerted of them via the web interface. This is an initial step towards enhancing and automating the upgrade process.
- Force Java. Previously ZoneMinder could be forced to override it's detection of browser capabilities to prevent the Cambozola Java applet being used. However sometimes the opposite effect was desired and using the applet was preferred to native image handling. This has now been made possible by making the ZM\_CAN\_STREAM option tri-state allowing 'auto', 'yes' or 'no' to be used to provide all alternatives.
- Alarms Cleared on Exit. In previous versions if an alarm was present when the analysis daemon (zma) exited the alarm would remain flagged. This had little effect except if the monitor was being watched however it was a bit annoying so any alarm flag is now cleared when this daemon exits.
- New Languages. Translations for Japanese (ja\_jp), French (fr\_fr) and Russian (ru\_ru) are now included.

### **10.13. Release 1.17.1**

Bugfixes and additional languages.

- Login Bug. A bug was identified whereby an unauthorised user could gain access to the console view of ZoneMinder. This was the only view available and no access to any camera views or configuration was possible. This bug is now fixed.
- New Languages. Two new language files were added. These allow ZoneMinder to use the German (de\_de) and Polish (pl\_pl) languages.
- Language File Format. The format of the language file was changed to allow the specification of character set and locale as well as have more flexibility in the calculation of plural forms.
- Option Language. The prompts and help text for the options is now also available for translation. A guide is included in the language file to allow this if necessary. Currently language translations exclude the options settings as this is a rarely accessed area and contains a great deal of text. The new format allows individual options to be translated piecemeal as the opportunity arises.

### **10.14. Release 1.17.0**

Language changes and other enhancements.

- Version Numbering. ZoneMinder version numbers have now changed. This is to allow more frequent 'point' releases which are expected to happen for instance whenever new language files are included. Previously all releases had the same version increment so it was difficult to tell the significance of any particular release. Now the version number is in the x.y.z format where a change in x signifies a major fundamental or architectural rework, a change in y will indicate a new release containing incremental feature changes or fixes recommend to all users and a change in z will generally mean minor non-functional or critical modifications which would not be recommended as important to all users. As ZoneMinder has been

referred to by the point release up until now, e.g. .15, .16 etc the next number in that sequence has been retained for continuity and to avoid having any ambiguity in version numbers.

- Language Support. ZoneMinder now allows specification of system and user specific languages other than UK English. These languages are given in language files named `zm_lang_nn_mm.php` which can be created from the default `zm_lang_en_gb.php` file. If your language is not included then please consider doing a translation by checking this file and submitting your changes back for inclusion in future releases.
- Syntactic Improvements. Previously setting 'NOTICE' errors on in PHP would flag tens or hundreds of violations in the ZoneMinder web files. Whilst not strictly errors this represented sloppy coding and sometimes covered up genuine bugs. All the files have been revisited and revised to ensure that a many of these problems as possible have been eliminated and only the very few where the fix would be significantly less optimal than the problem remain.
- Stream Scaling Resizing. Previously when watching a stream and modifying the scale of the streamed feed only the actual feed would change size and the containing frames and windows would remain the same. This was fine for changes to smaller scales but problematic for larger scales. This has been changed for that the window and frames will now resize appropriately.
- Mmap Return Value. A problem identified by users in the forum relating to checking of return values from the mmap function call has been corrected.
- Minor Bugs. A number of minor bugs and inconsistencies were corrected.

## **10.15. Release 0.9.16**

Major usability enhancement and fixes.

- Run States. Instead of the old 'start/stop' links the current system state is now a link which takes you to a dialog which allows you to start, restart or stop the system. You can also save the current run state which basically takes a snapshot of the current monitor functions and saves that. You can then reselect that state later which basically involves resetting the monitors to have these saved functions and then doing a system restart.
- New Monitor Functions. Instead of Passive, Active, and X10, the modes are now Monitor (= old Passive) which just allows you to watch the feed, Modect (= old Active) which is MOtion DetECT and which will capture events as previously, Record which continuously records with no analysis and MoCord which is a hybrid of Modect and Record and which will continuously record but also do motion detection and highlight where this has occurred. The Record and Mocord functions both records events whose length in seconds is defined by the 'Section Length' monitor attribute. You can additionally specify a 'Frame Skip' value to tell it to not record 'n' frames at a time, when not alarmed.
- X10 Function removed. The X10 mode has been removed and replaced by an indication of whether the monitor is 'continuous' or 'triggered'. This basically just indicates whether it may be controlled outside of `zmdc` and `zmpkg`. Additionally the X10 triggers may now be specified in an X10 section. This has changed to allow for other types of triggers to be added more easily.
- Paginated Event listings. The event listings are paginated by default. You can list all of the events if you like by choosing the appropriate option. There are shortcuts to pages of events at the top of the listing. If these produce strange looking sequences like 1,2, 3, 5, 9, 17, 37 etc this is deliberate and uses an exponential algorithm intended to allow you to quickly navigate through the list to a particular page in the minimum number of clicks.
- Scaleable Streams. Event and monitor streams can now be scaled to a certain extent allowing you to view at a different resolution than that captured. This area may be somewhat incomplete especially in terms of monitoring at a higher screen size where the frame is not adjusted properly.

- Variable Frame Rates. Event streams can now be viewed at various rates allowing faster review (if your bandwidth allows) to long events, or slower for more precision.
- Scaleable/Variable MPEG generation. Generation of MPEG videos now also allows you to specify the scale relative to the original image and also the frame rate. Again, for long events captured in the perpetual recording modes this will allow a faster review of the period the event covers.
- Tabbed Monitor options. Specification and modification of monitors is now in a tabbed form for easier navigation.
- Additional stream headers. The stream headers have been changed to hopefully ensure that they are less likely to be cached.
- Maximum process restart delay. `zmdc.pl` now has an upper limit (15 minutes) to the time it waits before restarting continuously crashing processes.
- Intelligent Module inclusion. `zmfilter.pl` now includes `Archive::Zip` and other modules on an as needed basis so won't complain about them being missing unless they have been explicitly configured to be used.
- Adaptive Watchdog. `zmwatch` now more adaptive to actual frame rates.
- Fixed `zmfilter` CPU sucking bug. `zmfilter.pl` will now restart on failure to read shared memory. Previously this could go into a nasty CPU sucking loop!
- New `zmconfig` options. `zmconfig.pl` has a new option to run with no database if necessary
- File reorganisation. Various administrative file changes and reorganisations.
- Compiler warnings. Various tweaks and modifications to reduce compiler and memory warnings.
- SQL Buffer size. Increased SQL buffer size to cope with large pre-event buffers, plus a couple of other buffers have been enlarged.
- Incorrect Frame time offsets. The time offsets in alarmed frames were incorrect and based on the time of storage rather than capture. This gave the impression that there was a delay after the first alarmed frame and messed up some streaming timings. This has been fixed.
- Event Frame listing. You can now view details of the frames captured such as their time and score etc by clicking on the scores in the events views.
- Refined shared memory handling. Fixed `zmfilter`, `zmwatch` and `zmx10` to allow zero as a valid shared memory id to allow them to keep on working if the segment has been marked for deletion
- Frame daemon stability. Changed image buffer in `zmf` to be static rather than dynamic. This has made `zmf` much more stable.
- MPEG overwrite option. Fixed the 'Overwrite' checkbox in video generation to actually overwrite the video. Modded the page slightly also.
- Daemon control improved. Changing between monitor functions, e.g. `Modect`, `Mocord` etc now restarts the correct daemons.
- Improved time based filters. Filters that include time based clauses now get executed regardless of whether new events are being generated.
- Audit daemon started unconditionally. `zmaudit` is now started regardless of the setting of `FAST_DELETES` as `zmfilter` depends on it being there.
- Filtering more active. `zmfilter` is now started in 'Monitor' mode. It does not run in when monitors are completely off however.
- Stills paged. The stills view of events is now paginated for easier navigation.

- Archive images optional. Normally when an alarm is detected a set of raw images is saved along with a mirror set of images containing motion highlighting. This second set can now optionally be disabled.
- Settings in auth mode. Control of camera brightness, contrast etc did not previously work when running in authorised mode. This is now fixed.
- zms parameter bug fixed. The streaming server incorrectly parsed and assigned one of it's arguments. This is now fixed.
- zmu brighness bug. Previously camera brightness was not correctly parsed from command line options passed to zmu.
- Event window width variable. Event windows now scale to fit the event image size.

## **10.16. Release 0.9.15**

Various bug fixes from the last release and before.

- Bandwidth. A bug was introduced in .14 which caused a corrupted console display and manic refreshes on new installations. This was due to a missing bandwidth setting when no existing cookie was detected. This is now fixed.
- Again in .14 a problem occurred for a new release whereby zmconfig wanted to know the database details and but also previously wanted to access the database before it had asked the questions. This has now been addressed though it does require that zmconfig is run twice initially, once to created the scripts and once to import the configuration into the database.
- In association with the previous error, the zm\_config\_defines.h file was not created in the absence of the database as this was what was used to assign configuration ids. This now takes place regardless of the database.
- The SQL to create the Users table was mistakenly omitted from the .12 database upgrade script this has now been corrected.
- A bug in zmfILTER was pointed out whereby the dynamic loading of the Zip or Tar archive modules depending on a preference actually wasn't. It was looking for both and loading both at compile time. This has now been modified to be fully runtime.
- The database user definitions in the zmvideo script indicated one database user while the database connection used a different one. This prevented any videos being generated.
- A problem was found if using the zmf frame server and greyscale images. The option to colourise JPEG images is intended to be used to ensure that all JPEG files are written with a 24 bit colourspace as certain tools such as ffmpeg require this. However in the circumstances described above images written by zma directly were colourised whereas those written by zmf weren't. A change has been made whereby if set all greyscale JPEG images are colourised in all circumstances.

## **10.17. Release 0.9.14**

Major new feature and important bug-fixes.

- Web configuration. Following many requests and to make ZoneMinder easier to administer the configuration system has been changed slightly. You should now still run zmconfig.pl to specify an initial configuration but you now only need to answer the first few questions to give a core set of options including the database options. The remainder of configuration options can then be ignored to start with and all but the core options will be written to the database. You can then view and modify these options from the web interface and apply then without recompilation, which is now only necessary if you change the core configuration.
- Following a number of requests the .sock file directory is now configurable in zmconfig.

- Y channel bug. When using colour cameras a motion detection problem was present if fast RGB images deltas (ZM\_FAST\_RGB\_DIFFS) was off. This was an overflow error in the calculation of the Y channel and caused excessive image differences to be calculated. This has now been fixed.
- The use of the Term::Readkey perl module in zmaudit.pl has been removed. This module had been removed from zmconfig.pl previously but had lingered in this script.
- A bug was found in zmx10.pl causing a crash if time delayed X10 events were used. This has now been fixed.
- Removed use of 'zmu' binary from zmwatch.pl and zmx10.pl. Previously these scripts had used zmu to determine last image time and alarm state information. The use of this script was a bit overkill and the introduction of user permissions complicated matter slightly so these scripts now access the shared memory directly.
- Shared memory permissions. Following introduction of a user permissions system the previous 777 mode for shared memory was deemed insecure. Consequently from now on shared memory is only accessible from the owner. This means that zmu will only work when run as root or the web user unless you set it setuid where it should still be secure as it will require authentication.
- All SQL buffers in the C++ code have been enlarged. There was previously an issue with a buffer overflow on certain occasions.

## **10.18. Release 0.9.13**

Beta version of several features and fixes, never generally released.

- Following a number of requests the .sock file directory is now configurable in zmconfig.
- Changed some of the core video calls to be V4L2 compatible. This primarily involved opening the video devices and memory maps as read/write and not just read-only.
- Shared memory has now been rationalised to prevent some common problems. Remember to shutdown the whole ZM package before installing, from this version on it will remove all old shared memory segments.
- Fixed not numeric comparison in zmwatch which was causing, or appeared to be causing, some errors.
- Fixed zone image map bug for percentage zones. When you had defined a zone in percentage terms, the image map used to select it for editing was broken. This is now fixed.
- New contrast/brightness etc adjustments feature. This accessible from the Settings link on the monitor window. It's fairly basic at present but should work for most types of cameras. If you have any device or driver specific auto-brightness, auto-contrast etc enabled the changes you make may appear to work but may be overridden by the auto feature immediately so check for that if your changes do not appear to be having an effect. Also if you have a number of cameras being multiplexed onto one device then any changes here will probably affect all your cameras.
- Some redundant window scrollbars removed.
- Added user and access control. If enabled in config (ZM\_OPT\_USE\_AUTH) then you will need to define and login as ZM users. There will be one users already defined, with username 'admin' and password 'admin'. This user is defined will full permissions and clicking on the 'Options' link from the main console window will allow you to modify and create users with various permission sets which hopefully will satisfy most requirements. These users (except any defined with 'system' edit capability) can be restricted to certain cameras by adding the monitor ids as a comma-separated list (no spaces) to the appropriate field. Users limited to specific monitors may not create or delete monitors even if defined with monitor edit permissions.



- Some windows now (optionally) use a JavaScript timeout to refresh themselves rather than a refresh header. Since refresh headers were interrupted if a link of the page was linked there were previously various forced refreshes from child windows to restart the refresh process. By using JS refresh timers which are not interrupted these extraneous refreshes have been mostly eliminated.

## **10.19. Release 0.9.12**

Mostly bug-fixes with a couple of minor features.

- Double first images. Fixed a problem where the first image of an event was being recorded twice. I don't think this was at the cost of any of the other images but one copy was an extra.
- Made zmdc connect more intelligent. On the suggestion of a couple of people I have made the zmdc.pl server spawning and waiting a bit more intelligent. Rather than waiting a fixed (short) amount of time, it now polls every second for a while, stopping if the connection is made. Thanks to Todd McAnally for the initial suggestion.
- Added image view to events lists. Again a partial implementation of a suggested feature. If you click on the score column you will now get a snapshot of the event frame with the highest score. This is to enable you to quickly see what the event was about without having to watch the stream or view all the static images.
- Make delta times variable precision. A couple of problems had been reported where long events got negative durations. This was due to an overflow in a time difference routine. This had been operating on fixed precision allowing high precision for short deltas. This routine has been changed to allow variable precision and events will now have to be several days long to wrap in this way.
- Fixed round detection problem. Although the existence or otherwise of the 'round' function is correctly detected, the appropriate header file with the results of this test was not included which was not helpful. This has been corrected.
- Fixed monitor rename bug. Renaming a monitor did not correctly modify the events directory to reflect this. This has now been fixed.
- OPT\_MPEG bug. A bug was reported (by Fernando Diaz) where the results of the ZM\_OPT\_MPEG configuration variable was not correctly imported into the scripts. This now happens as intended.
- Fixed zmvideo.pl event length bug. The zmvideo.pl script which is used to generate video MPEG files tries to calculate the correct frame rate based on the length of the event and the number of frames it contains. Previously it did not take account of the pre and post event frames and so passed a much shorter value to the mpeg encoder than it should. This will only have affected short events encoded with ffmpeg but will have resulted in much faster frame rates than necessary. This has now been corrected to take the whole event length into account.
- Fixed remote camera memory leak. A memory leak was reported when capturing with remote cameras, this is now fixed.
- Orientation. Added option to rotate or invert captured images for cameras mounted at unusual angles.
- Fixed filter bug. A bug in the zmfilter.pl script was detected and reported by Ernst Lehmann. This bug basically meant that events were not checked as often as they should have been and many may have been left out for filters that had no time component. The script has now been updated to reflect Ernst's suggested changes.
- Stylesheet change. Previously the stylesheet didn't really work very well on Mozilla, Netscape and browsers other than IE. This turned out to be because I was using HTML style comments in there instead of C style ones. This has now been corrected so you should see the correct styles.

- Zmconfig.pl ReadKey. Thanks to a ridiculously sensible suggestion from Carlton Thomas this module has been removed from zmconfig.pl. Originally Term::ReadKey was in there for funky single character unbuffered input but that has long since disappeared so just regular perl input methods are used now. This removes one of the most irritating features about ZoneMinder installs.
- Delete monitor confirm. Due to some unfortunate accidents by users, attempts to delete monitors will now require confirmation.
- Detect linmysqlclient.a. Added better detection script into 'configure' top spot when libmysqlclient.a is missing.

## **10.20. Release 0.9.11**

Various new features and fixes.

- Added stats view – If you have the RECORD\_EVENT\_STATS directive set and are viewing a still image from an event you can now view the statistics recorded for that frame. This tells you why that frame triggered or participated in an alarm. This can be useful in tuning the various motion detection parameters and seeing why events occurred.
- Tabulated events – The main events view is now tabulated to look a bit nicer.
- New video palette support – As well as the existing greyscale and 24 bit RGB palettes, you can now choose YUV420P and RGB565. Rewrote the palette/colours area a bit to enable support for other palettes in the future if requested. Bear in mind though that YUV palettes are converted into RGB internally so if you have the choice RGB24 may be faster as it's the 'native' format used within.
- Added preclusive zones – Added a new zone type, the preclusive zone. For full details see the relevant section above but in brief this is a zone type that if alarmed will actually prevent an alarm. This completes the pantheon of zone types I think.
- Fixed Mozilla JavaScript – Various JavaScript functionality did not function on Mozilla, Netscape and other browsers. This is now (hopefully) fixed.
- Allow image and mpegs to be attached to emails – Added new tokens (%E1%, %EIM% and %EV%) to the filter emails. This allows the first alarm image, most highly scored alarm image and an alarm MPEG to be attached to alarm notification emails. Use %EV% especially with care!
- Fixed possible motion detection bug – I found a few double declared local variables left over from the rewrite. This may have affected the motion detection algorithm. Fixed now anyway.
- Modified scoring – Alarm scoring has been modified to give more granularity for smaller events. This will have the effect of raising the scores for small events while large ones will still be about the same.
- Fixed /cgi-bin path problem – Previously you could specify the real path to you cgi-bin directory if you have one but not the web path. You can now do both.
- Improved video handling in browser – The MPEG/video area of the web GUI had been a bit neglected and looked somewhat ugly. This has now been improved to a degree and looks a bit nicer.
- Added ffmpeg support – Historically ZoneMinder has only supported the Berkeley mpeg encoder which was slow and rather limited. ZoneMinder now supports the ffmpeg encoder as well which is much much faster and makes generation of MPEG videos at realistic frame rates more of a reality. As ffmpeg has so many options and everyone will probably want a different emphasis you can now also specify additional ffmpeg options via zmconfig.pl.
- Colourise greyscale image files – In past versions, captured greyscale images were stored as JPEG files with a corresponding greyscale colourspace. This saved a small amount of space but meant that mpeg\_encode had to do a conversion to encode them, and ffmpeg just fell in a heap. Now you can optionally opt to have greyscale images saved as full 24 bit

colourspace images (they still look the same) at the price of a small penalty in CPU and disk but allowing you to easily and quickly create MPEG files. This option is one by default but can be switched off if you do not require any MPEG encoding.

- Fast RGB diffs – Previously ZoneMinder used quite a loose method for calculating the differences between two colour images. This was basically averaging the differences between each of the RGB components to get an overall difference. This is still the default but by setting ZM\_FAST\_RGB\_DIFFS to 'no' you can now make it calculate the Y (or brightness value) of the pixels and use the difference between those instead. This will be more accurate and responsive to changes but is may be slower especially on old machines. There is a slight double whammy here if you have a YUV palette for capture and set this option off as the image will be converted to RGB and then partially converted back to get the Y value. This is currently very inefficient and needs to be optimised.
- Fixed STRICT\_VIDEO\_CONFIG – Previously this actually behaved the opposite of what it was supposed to, ie. if you wanted it strict it wasn't and vice versa. Thanks to Dan Merillat for pointing this one out.
- Web colour change – I thought the old red, green and amber text colours were just a bit too gaudy so I've toned them down a bit. Hope you like them!

## **10.21. Release 0.9.10**

Many bug-fixes and major feature enhancements.

- Configure 'round' bug - Fixed a problem with the configure script that didn't detect if the 'round' function was already declared before try to do it itself.
- Low event id bug - Fixed bug where events with an id of < 1000 were being cleaned up by zmaudit.pl by mistake.
- Source file restructuring - The source files have been broken up and renamed extensively to support the first stage of the code being straightened out. Likewise the class structure has been rationalised somewhat. The php file names have also changed in some cases so it might be best to delete all your php and css files from the zone minder install directory first as the old ones won't be overwritten and will be left behind.
- Streamed cycle view - The monitor cycle view (the one where each monitor is displayed sequentially) now supports streams as well as stills.
- New 'montage' view - Added a montage view showing all your cameras simultaneously either streaming or stills. The width of this window (in terms of number of monitors) is a configuration option.
- Network camera support - A major change in this version is support for remote or network cameras. This is currently implemented as series of http grabs of stills rather than being able to break up motion jpeg streams. However frame rates of from 2-10 should be achievable depending on your network proximity to the cameras.
- Option BGR->RGB swap - Added the option to switch on or off the inversion of RGB to BGR for local cameras. It is on by default to maintain compatibility with previous releases.
- zmu suspend alarm option - Added new -n option to zmu to effectively suspend alarm detection for a monitor. This is intended for short term use and to support PTZ cameras where alarm detection is desired to be suspended while the camera changes orientation or zoom level.
- FPS limiting - Added a new option to monitors to add a maximum capture rate. This allows you to limit the amount of hits a network camera gets or to reduce the system load with many cameras. It also works with multi-port cards and limiting the capture rate on one camera allows the spare FPS to be allocated to other devices. For instance with two cameras and no throttle, I get about 4FPS each. Throttling one to 2FPS allows the other to operate at 6FPS so you can allocate your capture resources accordingly. This limiting can be disabled while alarms are occurring as a global option in zmconfig.pl.

- Alarm reference update - Added option to not blend alarmed images into the reference image. See the help in zmconfig.pl for caveats.
- Disappearing monitors - Fixed the disappearing monitor problem in the console view where monitors with no events were randomly not being shown.
- Clean and tidy - Cleaned up a load of compiler warnings and miscellanea to ensure a cleaner happier build.
- Streamed image headers - Made all headers in streamed images have full CRLF termination which will hopefully now prevent the problems with broken streams that had existed mostly with Mozilla (and hopefully won't break anything else).
- Expire streams - Added expiry headers to streamed images so they will always display fully.
- Event navigation - Added next, prev, delete & next, delete & prev navigation to events to allow you to quickly review events in sequence as had been requested by a number of people.
- USR blocking – The debug USR signals were not being blocked properly leading to nasty effects in zmc mostly.
- zmfiler execution – Previously zmfiler execution was not synchronised with the monitor state or the analysis daemon leading to it sometimes being run unnecessarily. From now on the zmfiler process will only run when a monitor is active and so actually potentially generating alarms.
- zmdc short statuses – Removed the logging of the short status values that zmdc.pl returns to it's clients which had been clogging up the log file.
- Bugs and pieces - Fixed various bug(ettes) that I came across that that I don't think had been reported or noticed so I don't think we need to talk about them here do we.

## **10.22. Release 0.9.9**

Mainly bug-fixes and minor feature enhancements.

- Added zmu -q/--query option - There is now a new query option for zmu. When combined with -d it gives the config of the device and when used with -m it dumps the current settings for the monitor and zones. Mostly useful for bug reporting. The previous version of zmu used with just -d gave this information for a video device by default. This now requires the -q option also to bring into line with it's -m equivalent.
- Added creation of events directory - Previously the 'events' directory was not created on install, this has been fixed.
- Can now retag PHP files if necessary - Version 0.9.8 was the first version to use short\_open\_tags in the PHP files. This caused grief to some people so this script will put them back to the long verion.
- Frame and event lengths fractional - A new field has been added to the Frames table. This is 'Delta' and is a fractional number of seconds relative to the event start time. This is intended to support the real-time playback of events rather than just 'as fast as possible' or with a configured delay as at present. The event length is now also fractional.
- Corrected extraneous Width to be Height - The last version of zmu included a Width comment which should have been height.
- Changed colour depth to bits - Having colour depths expressed in bytes has caused no end of problems. This is now changed to be bits and can be changed via a dropdown to limit what can be entered. Don't forget to run the zmalter script to update your DB.
- Renamed terminate to zm\_terminate - The use of 'terminate' in zmc.cpp caused a conflict on some systems so renamed it to something more specific.

- Zone deletion problem - A problem was found such that when deleting zones the appropriate daemons were not being asked to restart daemons correctly.
- Console changes - The current version number is now displayed in the console. A refresh button has also been added along with a minor reorg.
- Added delete button enable to checkAll - Using the 'Check All' button in the main monitor window previously did not enable the delete button. This is now fixed.
- Reload on click - In previous versions the console window would reload if a monitor window for example was clicked. This was removed in the last version which meant that sometimes the console never got refreshed as its timing loop was broken. This functionality has now been reinstated.

### **10.23. Release 0.9.8**

Several new features and bug-fixes

- Upgrade note - If you have installed 0.9.7 and wish to save your configuration then copy your existing zmconfig.txt file over to your 0.9.8 directory and before running zmconfig.pl.
- Added multiple options to zmu - You can now give multiple options to zmu and get all the responses at once. However this is currently in a deterministic order and not related to the order you give them.
- Added -v/--verbose option to zmu - Zmu has been made more human friendly though it still remains primarily for daemon use. Giving the -v or --verbose option prints out a bit more as a response to each command.
- Add -d/--device to zmu - This option is designed to allow you to get your video device working with another application such as xawtv and then use zmu -d to print out the settings it's using
- (especially with the -v option). These options can then be used as a starting point for your ZoneMinder configuration.
- Added FPS in status field - The status field in the web monitor views now contains an FPS setting as well as the status.
- Zmconfig changes - zmconfig handles missing options better and rewrites config file even in non-interactive mode.
- Fixed config problems in zmcfg.h - Some config was not being set up correctly in zmcfg.h.
- Zmwatch now works on image delay and not fps - Previously the zmwatch daemon detected capture daemon failure by trying to use the FPS setting. This was imprecise and prone to false readings. It now uses the time delay since the last captured image.
- Added zmpkg.pl and zm scripts - There are now two new scripts. zmpkg.pl is in charge of starting and stopping ZoneMinder as a whole package and zm is designed to be (optionally) installed into your init.d directory to use ZoneMinder as a service.
- Fixed bug in Scan mode - The monitor cycle or scan mode had stopped working properly due to images not being generated. This is now fixed.
- Revamped the console window slightly - The console window has now been reformatted slightly to give more and better information including server load.
- Added email and messaging to filters - Filters now allow you to send emails or messages (basically just short emails intended for mobile devices) on alarms. The format and possible content for these emails is in zmconfig\_eml.txt and zmconfig\_msg.txt.
- Made zmdc more aggressive in killing old processes - The zmdc.pl daemon will now kill any ZoneMinder processes it finds on startup or shutdown to prevent orphans from being left around.

- Configuration changes - Previously there were a lot of files generated by configure. Now only zmconfig.pl is generated this way and all the other configuration files are created by zmconfig.pl (from .z files) to centralise configuration more.
- Fixed cambolzola opt bug - There was a bug in the Cambozola options, I can't remember what it was but it's fixed!
- Retaint arguments in zmdc.pl - In some installations zmdc was complaining about tainted arguments from the socket. These are now detainted prior to sending and after receiving.
- Forced alarms - You can now force alarms when looking at the monitor window should anything catch your attention. You have to remember to switch them off as well though.
- Looser video configuration - Some video configuration errors can now be ignored via the STRICT\_VIDEO\_CONFIG option.
- Monitor window refresh on alarm - When the monitor window is active and an alarm has occurred the most recent alarms list is immediately refreshed to show it.

## **10.24. Release 0.9.7**

Yes, a big jump in release number but a lot of changes too. Now somewhat more mature, not really an alpha any more, and a lot of bugs fixed too.

- Added zmconfig.pl script to help with configuration.
- Revamped to work better with configure scripts
- Monitors now have more configuration options, including some that were statically defined before such as location and format of the image timestamps.
- Removed Alarms table from schema as not required, never was actually...
- Added a number of new scripts, see the scripts directory
- Added Fast delete to PHP files. This allows the web interface to only delete the event entries themselves for speed and then have the zmaudit script periodically tidy up the rest.
- Added event filter to enable bulk viewing, upload or deletion of events according to various attributes. Filter can be saved and edited.
- Added last event id to shared memory for auto-filtering etc.
- Changed zmu -i option to write to monitor named image file.
- Made shared memory management somewhat more sensible.
- Now stores DB times as localtime rather than UTC avoiding daylight saving related bugs.
- Fixed bug with inactive zones and added more debug.
- Changed main functions to return int.
- Added help and usage to zmu.
- Fixed browser acceptance problem, more easily defaults to HTML.
- Split out the PHP files into a bunch with specific functions rather than one monolithic one.
- Fixed NetPBM paths and changed \_SERVER to HTTP\_SERVER\_VARS.
- Added HUP signal on zone deletion.
- Added NETPBM\_DIR and conditional netpbm stuff.
- Removed hard coded window sizes, all popup window dimensions can be specified in zmconfig.php

- Changed form methods to 'get' from 'post' to avoid resubmit warnings all the time.
- Added conditional sound to alarm on web interface.
- Fixed syntax error when adding default monitor.
- Some of the web views have changed slightly to accommodate the separate events view.
- And much much more, probably...

### **10.25. Release 0.0.1**

Initial release, therefore nothing new.

## 11. To Do

Seeing as ZoneMinder is so young and has kind of evolved rather than being planned there are a bunch of improvements and enhancements still to do, here is just a sample.

- Perhaps split out devices - I think devices should probably be a separate table and class from monitors. Not critical but would represent a better model.
- Comments - Needs many more, but that's just me I'm hopeless at commenting things out. I'll get round to it soon though honest! You're lucky to even get this document.
- Optimised zones - The zones could do with being sorted out a bit to optimise the processing of overlapping ones, at the moment you can waste resource unless your zones are kept very tidy.
- Create zones using server side image maps - This would make it easier to precisely define and see where your zone is going to go. Not critical but handy but a bugger to do.
- Zone Definitions - Allow zones to be defined according to a colour coded bitmap or as polygons. Currently all zones are rectangular this would add a bit of flexibility. Would need a bit of a rewrite though. This will incur a slight penalty on startup and a very slight one on processing for all reasonably shaped zones. Work has already begun on this feature.
- Mouseover help - A bit more help popping up when you mouseover things would be handy. A bit more help full stop actually.
- Automatic device configuration - Video 4 Linux supports various device queries, it should be possible to get most of the device capability information from the device itself. The zmu utility does this now but it's not yet integrated into the web pages.
- Extend the API. Well ok it's not really got an API yet but the image data is held in shared memory in a very simple format. In theory you could use the capture daemon to gab the images and other things could read them from memory or the analysis daemon could read images from elsewhere. Either way this should be done through an API, and would need a library I think. Also the zmu utility could probably do a whole lot more to enable other things to manage when the daemons become active etc.
- Allow ZoneMinder to 'train' itself by allowing the user to select events that are considered important and to discard those that should be ignored. ZoneMinder will interpolate, add a bit of magic, and recommend settings that will support this selection automatically thereafter. The hooks for this are already in to some extent.
- Add sound support to allow a captured audio channel to be associated with a video device. Work on this feature has already begun.



## 12. Bugs

- When opening a link to an event etc from a notification email the window that is opened is just a regular browser window and not in the context of a proper ZoneMinder web interface. Thus it comes up too big usually (not a major issue) and also things like 'Delete' don't work as it wants to do things to its parent (which is more of a major issue).
- The .sock files used by the \*nix sockets I suspect may have the odd permission issue now and again. I think everything recovers from it but it needs checking out.

Probably bucket loads more, just fire them at me.

### 13. Non-Bugs

- Yes, those are tabs in the indents; I like tabs so don't go changing them to spaces please. Also, yes I like my opening braces on their own line most of the time, what's the point of brackets that don't line up?

Everything else that isn't definitely broken is probably deliberate, or was once anyway.

## 14. License

ZoneMinder is released under the GPL, see below.

ZoneMinder README, \$Date: 2004/01/13 22:12:43 \$, \$Revision: 1.9 \$

Copyright (C) 2004 Philip Coombes

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.