

# Location and configuration files for VoipNow logs

Applies to VoipNow 4.0.0!

This article describes the logs generated by VoipNow components and provides the methods to configure them.

## Step-by-step guide

### VoipNow web interface

To see what requests are being performed on the web interface, or if you have errors, check the following logs:

```
/usr/local/voipnow/admin/log/access.log
/usr/local/voipnow/admin/log/error.log
```

### Configuration

The configuration file is `/usr/local/voipnow/admin/conf/voipnow.conf`

To change the debug level, check the following lines:

```
# Error handling
access_log /usr/local/voipnow/admin/log/access.log main;
error_log /usr/local/voipnow/admin/log/error.log notice;
```

where `error_log` can be `[debug | info | notice | warn | error | crit | alert | emerg]`

### PHP

This log is useful in case of web interface issues.

You can find the log file here:

```
/usr/local/voipnow/admin/log/php_error.log
```

The loglevel in this file is defined by the level set in table `hs_audit_zone` (default 3, maximum 7)

### Kamailio

The Kamailio log is useful in case you have problems with calls, registration or any other call-related issue, except for audio issues.

You can find the log here:

```
/var/log/kamailio/kamailio.log
```

### Configuration

The configuration file is in `/etc/kamailio/kamailio.cfg`

```
debug=-1
facility_sql=no
facility_sipmsg=no
facility_hr=no
```

where `debug` can have values from -1 to 9.

You can also enable various facilities to add extra information to the logs. For example, `facility_sipmsg` will include the SIP packets in the log.

All changes here require a Kamailio restart.

For example, the debug level can be set to level 3 example with the following command:

```
kamctl fifo debug 3
```

The change immediately becomes active and no restart is required.

## Asterisk

The Asterisk log is useful in case of call-related issues. You can find the log files here:

```
/var/log/asterisk/messages
```

Or directly in the console, using the following command:

```
asterisk -rcvvvvvvvvvvvvvvv
```

## Configuration

The content of the log file or of the console is configurable in `/etc/asterisk/logger.conf`:

```
console => notice,warning,error ; what to log on console (add "fax" to log the fax error messages as well)
messages => notice,warning,error      ; what to log on messages
```

Extra flags like `moh`, `fax`, `sip` and `rtp` are available. `LogLevel` can be configured from the asterisk cli.

For example, to set debug and verbose to level 1111, run the following command:

```
asterisk -rx "core set debug 1111"
asterisk -rx "core set verbose 1111"
```

These commands will only take effect with debug and verbose active.

## Hubring

You can find the Hubring log here:

```
/var/log/hubring/hubring.log
```

It is recommended to troubleshoot service issues - for example, if Hubring does not start.

To see all operations performed with Hubring key, run the following command:

```
hr-cli -dp <password> -n 10 monitor
```

where `password` is the `dbkey` value from `/etc/hubring/hubring.conf`)

## Database

The log files created by MariaDB are the following:

```
/var/lib/mysql/<hostname>.err #mainly startup/shutdown messages
/var/lib/mysql/<hostname>.log  #queries ran in mysql by all applications
```

To see all queries performed in the database, in the log file, run the following command:

```
mysql -u`cat /etc/voipnow/.sqldb` | awk -F ":" '{print $2}' -p`cat /etc/voipnow/.sqldb | awk -F ":" '{print $3}'` voipnow -e 'set global general_log=ON;'
```

To disable the log, run the same command, but use `general_log=OFF` instead of `general_log=ON`. This will log all queries performed in MySQL in the `/var/lib/mysql/<hostname>.log` file.

## Postfix

To troubleshoot mailing issues, use the Postfix log file located here:

```
/var/log/maillog
```

## Related articles

- [How to detect abnormal traffic using Pike](#)
- [How to use Request Logs](#)
- [Troubleshooting fax session failures caused by incorrect negotiation of T38FaxMaxDatagram parameter](#)
- [Troubleshooting calls and debug steps](#)
- [How to use SIP trunking to connect a PBX to an extension](#)