up.time Diagnosis

up.time's logs can assist you with diagnostic steps that you may need to perform should you encounter problems. Problem reports can be generated for up.time Customer Support if further analysis is required.

All up.time logs are written to the `/logs` directory, and problem reports to the `/GUI` directory, both of which are found in the up.time installation directory:

- Linux: `/usr/local/uptime/`
- Windows: `C:\Program Files\uptime software\uptime`

Topics on this page

Logs

The following logs are available for troubleshooting. Depending on the type of investigation, output from multiple logs can be correlated.

Log Name	Description and Uses	uptime.conf parameter and values
`uptime.log`	This is the base up.time log. System events are automatically recorded to these weekly logs, which follow the `uptime.log.<year>-<week>.log` naming format. You can determine the type of system information up.time writes to the log (ranging from verbose, to informational, to critical errors) by setting the logging level. The default setting, `INFO`, essentially logs all system event types that are higher than the service or thread level (which are logged at the `DEBUG` setting). To reduce the number of log entries, you can limit logging to events with a higher level of severity, from `WARN` to `FATAL`. Note that each severity level is a subset of higher levels (e.g., setting `loggingLevel` to `WARN` means any `WARN`-, `ERROR`- or `FATAL`-level events are written to the log).	`loggingLevel=` • `DEBUG` • `INFO` (default) • `WARN` • `ERROR` • `FATAL` • `ALL` • `OFF`
`uptime_diagnostics.log`	This log is similar to the `uptime.log`, but has a more detailed breakdown of system information to assist with troubleshooting. Additional information includes the name of associated thread, the name of the up.time component that logged the event, Element details, and if applicable, monitor, Element, VMware, user details. Like the `uptime.log`, the number of log entries is also set by the `loggingLevel` parameter.	`loggingLevel=` • `DEBUG` • `INFO` (default) • `WARN` • `ERROR` • `FATAL` • `ALL` • `OFF`
`uptime_exceptions.log`	All `DEBUG`-level Java runtime exceptions evoked by up.time actions. Full stack traces are channeled to this log to lighten and accompany the core `uptime.log` and `uptime_diagnostics.log` files. Use the context marker in the core log to find the exception in this log.	N/A
`uptime_console.log`	All Java-related command-line feedback based on up.time activity is routed to this log, providing extra information that may not be captured in the standard up.time log.	N/A

`audit.log`	up.time can record changes to the application's configuration in an audit log, and is essentially a record of which user performed which action, and when. The following is an example of an audit log entry: `2006-02-23 12:28:20,082 - kdawg: ADDSYSTEM [cfgcheck=true, port=9998, number=1, use-ssl=false, systemType=1, hostname=10.1.1.241, displayName=MailMain, systemSystemGroup=1, serviceGroup=, description=, systemSubtype=1]` There are many uses for the audit log. For example, you can use it to track changes to your up.time environment for compliance with your security or local policies. You can also use the audit log to debug problems that may have been introduced into your up.time installation by a specific configuration change; the audit log enables you to determine who made the change and when it took effect. By default, the `auditEnabled` parameter in `uptime.conf` is not defined, which means it is effectively disabled.	`auditEnabled=` • `yes` • `no`
`uptime_access.log`	A summary of which up.time access-related actions, mainly database queries, were evoked by which service or user, and the execution time. This database-focused log can be used in conjunction with the more user-focused `audit.log`.	N/A
`thirdparty.log`	Aggregation of warnings and errors logged by thirdparty libraries that up.time is using, such as the iReasoning library for SNMP monitoring. Correlating these with the other logs may help with investigation.	N/A
`uptime_sql.log` `uptime_sql_timing.log`	When SQL logging has been enabled with the assistance of uptime software Customer Support, these log shows all SQL queries, with and without execution time, respectively. Queries in `uptime_sql.log` are listed before execution, which can be compared with the second log to determine conflicts and deadlocks.	contact Customer Support

Problem Reports

When you encounter a problem with up.time, Customer Support needs a specific set of information to diagnose and fix the problem. up.time can automatically collect this information and compress it in an archive which you can send to Customer Support.

The archive contains the following:

- up.time configuration files
- system information
- log files
- database information and error files
- Java `hs_err_pid` error files
- a listing of the DataStore directory
- optionally, a copy of the configuration data from the DataStore

The archive is saved to the `GUI/problemreports` directory on the Monitoring Station and has a file name with the following format:

```
prYYYYMMDD-HHMMSS.zip
```

- `YYYYMMDD` is the date on which the report was generated (for example, `20101224`).
- `HHMMSS` is the time at which the report was generated (for example, `202306`).

Generating a Problem Report

To generate a problem report, do the following:

1. On the up.time tool bar, click **Config**.
2. In the tree panel, click **Problem Reporting**.
3. Configure the **Report Options**:
    a. Indicate whether to **Include configuration and service monitor status history**, and if so, how many months' worth of data.
    b. If configuration information is included, indicate whether to also **Include the last hour of performance data**.
    Adding performance data can result in a significantly larger problem report, requiring an appropriate amount of resources to generate, and time to download. This data, however, can help determine whether your up.time instance is running correctly.
    c. Indicate whether to include the **database check output** in the problem report.
    When this option is enabled, up.time runs the `dbchecker` script with the default values on your DataStore. This integrity test allows you to ensure there are no database issues that are part of, or are at the root of the problem. Disable this check box to improve generation performance by skipping the database check.
4. Click the **Generate Report** button.
    When the report has been generated, it will appear in the **Existing Problem Reports** section below, along with problem reports that have been previously generated.

5. Click the name of the problem report to download it to your local file system, then send the archive to uptime software Customer Support at support@uptimesoftware.com.