

Diagnosing Server Performance Issues

The following article serves as a diagnostics checklist to investigate server performance issues. If you are experiencing performance issues, review the sections below to diagnose the cause of the issue. If you are unable to identify the performance issue on your own, escalate the issue to our support team. For more information, see [Escalate an Issue](#).

Prerequisites

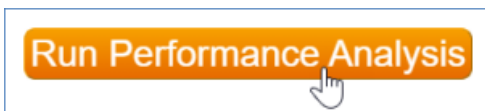
Depending on your release version and the method of investigating server performance, you may need the following:

- Access to the server file system
- Access to the admin console

Run Performance Analysis

Before completing any other steps to investigate server performance issues, run the performance analysis from the admin console by going to **Debugging > Performance** and clicking Run Performance Analysis. The analysis is designed to identify the specific activities that may be causing performance problems. It checks the CPU and memory utilization and produces a summary of events and activities for the day. The summary can help determine if an incorrectly configured rule or action should be investigated as a potential cause of performance issues.

The automated report opens in a new window, including sections for main activities, top CPU processes, and VMStat output. The report captures activities beginning at 12:01 a.m.



Run Performance Analysis button



If you need to run an analysis on a different time frame, or if your system is running a release earlier than 2019_01 r195525 dated 08/01/2019, you can run the script manually from the command line if you have direct access to the server. Download the script here: [performance_analysis.py](#) and run it with this command: `-sudo python performance_analysis.py`

You can also configure your server to gather and store performance data for GUI operations, so you can analyze performance over time. To turn on data gathering, go to General > Variables in the admin console and set the Save Performance Data variable to Yes. When this setting has been enabled long enough to gather data, ideally at least a month, you can go to Performance > Server Metrics or Performance > KB Metrics to see the data.

Diagnose Issues on a Self-hosted Server

The following sections are helpful to investigate server performance issues when you host Agiloft on your own server. If you use our hosted service, see [Escalate an Issue](#) below.

Verify the Hardware Configuration

Complete these steps to verify your hardware configuration:

1. Check your hardware configuration to confirm that it follows [Agiloft System Requirements](#) for hardware.
2. Run a performance test through the [Administrator Console](#) to validate that the hardware meets Agiloft standards. The test takes about ten minutes and strongly impacts performance while it's running, so it is best performed outside of business hours. See [Optimizing System Performance](#) for additional details on expected results of the performance test and more detail on other aspects of your system that can affect performance.

Check CPU and Memory Utilization

To investigate any CPU and memory utilization issues reported by the `performance_analysis.py` script, and to determine if a third-party application is taxing those resources, use the following techniques, depending on your operating system:

- For Linux servers, use the `uptime`, `top`, and `free` commands.
- For Windows servers, use the Task Manager or the Resource Monitor.

CPU and memory utilization regularly fluctuate. When investigating performance issues, look for applications that consistently consume CPU resources or RAM on a server.

If you determine that a third-party application is responsible for the high load or memory utilization, review the log files or thread dumps for that application.

Run Agiloft Diagnostics

JBoss is the primary process for the Agiloft application. On Linux servers, the process appears as `java`. On Windows servers, it appears as `ewjboss`. Running diagnostics on the process and reviewing log files can help identify any performance issues.

Thread Dump


If a performance issue occurs while executing a particular set of steps in the interface, repeat the steps and take a JBoss thread dump at the same time. To take a thread dump, run the following command on the server as root or an administrator, substituting the directory where Agiloft is installed for `[AGILOFT_HOME]`:

```
[AGILOFT_HOME]/bin/ew-control -d tr
```

Take two or three thread dumps and review them to isolate the cause of the performance issue.

Heap Dump

If the Agiloft logs show memory-related problems, such as "OutOfMemory" errors, then a heap dump may be necessary.

 Heap dumps are resource intensive and can take hours to complete. Run them outside of business hours or be ready to end the operation if it is severely affecting performance.

To perform a heap dump, run the following command on the server as root or an administrator, substituting the directory where Agiloft is installed for `[AGILOFT_HOME]`:

```
[AGILOFT_HOME]/bin/ew-control -d hp  
or  
[AGILOFT_HOME]/bin/ew-control -d all (sometimes the -d hp option fails but -d all succeeds)
```

Escalate an Issue

If the previous diagnostics do not reveal the cause of the performance issue, or if your system is hosted by Agiloft, escalate the issue to our support team for further investigation.

Submit Performance Issues for Further Investigation

Log in to our Support Portal and submit a support ticket with the following information:

- Steps to reproduce the issue.
- Knowledgebase credentials.
- Impact assessment, including the number of people the issue is affecting.
- Thread dumps and log files. If the server is hosted by Agiloft, grant access to the location of the logs and document the location in the ticket.