

Logs and Information Required in the Event of a QPS System Failure

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Capture Information](#)

[Diagnostics and Debug Logs](#)

[QPS License Information](#)

[System Statistics](#)

[Thread Configuration in Policy Builder](#)

[Fatal Error Log](#)

Introduction

This document describes the steps that must be completed in order to capture information when a Quantum Policy Suite (QPS) system failure or crash occurs. If the hardware, software, and virtual machine requirements are met, it is unlikely that the QPS will crash.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on these software and hardware versions:

- QPS Release 5.5 and later.

Note: Certain logs will not appear in QPS releases older than QPS Release 5.5.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Capture Information

If a QPS system failure happens, collect this information:

Diagnostics and Debug Logs

1. Log in to the Policy and Charging Rules Function (PCRF) client virtual machine (for example, pcrfclient01) and collect diagnostic information (for example, `/opt/broadhop/installer/diag/diagnostics.sh`).
2. Log in to the PCRF client virtual machine and collect debug information. Debug information includes consolidated QNS log, svn repo, and QNS configuration details. Make sure that the consolidated logs cover the time of the system failure and that the debug level is set in the `logback.xml` file.
3. Collect this output from your QPS (for example, Run `/opt/broadhop/installer/diag/zip_debug_info.sh` and the output is stored in `/var/tmp/debug_info<date>.zip`).

QPS License Information

1. Log in to the PCRF client virtual machine and collect QPS license information. A QPS is usually licensed for a specific feature and there is a maximum number of concurrent sessions it supports. The QPS also has an expiration date for this feature.
2. Navigate to this directory: `/etc/broadhop/license` and capture the output of the license (`.lic`) file. (for example, `cat /etc/broadhop/license/QUANTUM201311210402429360.lic`).

System Statistics

1. Capture the system statistics (Example: CPU, memory, disk utilization).
2. Log in to the PCRF client virtual machine and collect the output.
Example: `/opt/broadhop/control/top_qps.sh`
3. Log in to virtual machine that corresponds (for example, pcrfclient0x, lb0x, qns0x) and capture these system statistics:
cat /proc/meminfo > Allocated memory information
free -s 60 > Memory statistics for every single minute
vmstat 1 > CPU status for every single minute
ps -aux | head -10 > Top 10 process details which consumes most of CPU utilization
swapon -s > swap usage summary per device
. du -a | sort -n -r | head -n 10 > Top 10 files / directories consuming more space
4. Login to sessionmgr virtual machine and collect the outputs **mongostat** and **mongotop**, which will help in order to troubleshoot whether the issue is related to the database or not.

Thread Configuration in Policy Builder

Log in to policy builder and navigate to Reference Data > System-1 > Plugin Configurations > Threading Configuration.



The number of threads might range from 40 to 50 for TPS, but is less than 1,000. The maximum number of threads you can configure is 50. If you increase the number of threads, this impacts system performance.

Fatal Error Log

When a system failure occurs, the QPS generates a fatal error log, which contains the state of the process at the time the fatal error occurred. Fatal error or fatal exception errors cause the program to abort.

The fatal error log includes this information:

- The operating exception or signal that provoked the fatal error
- Version and configuration information
- Details on the thread that provoked the fatal error and the thread's stack trace
- The list of running threads and their state
- Summary information about the heap
- The list of native libraries loaded
- Command-line arguments
- Environment variables
- Details about the operating system (OS) and central processing unit (CPU)

The default log file name follows this format: **hs_err_pid<pid>.log** and is generated in the working directory where the corresponding Java processes started. Example: the working directory of the user when the user started the QNS process.

If you do not know the working directory, search the system for the file with the the name **hs_err_pid*.log** and examine the file for a time that matches when the error occurred.

Complete these steps in order to specify the location for the fatal error:

1. Log in to the pcrfclient01 virtual machine
2. Open jvm.conf (for example, vi /etc/broadhop/pcrf/jvm.conf).
3. Add the option: `-XX:ErrorFile=<directory>/<file-name>%p.log` to the list and make sure that the specified directory path exists and that the user QNS has full permission over that directory. **Example:** `-X:ErrorFile=/home/qns/fatal_error%p.log`
4. The "synconfig.sh" command can cause a lot of problems if the conf files in pcrfclient01:/etc/broadhop are not in synch with the conf files in /etc/broadhop on the VMs running the QNS service. The synconfig.sh will take the pcrfclient01:/etc/broadhop conf files and over write the conf files in /etc/broadhop on the VMs running the QNS. **Warning:** The command **synconfig.sh** will take the **pcrfclient01:/etc/broadhop conf** files and overwrite all the conf files in **/etc/broadhop** on the virtual machines running the QNS service (ifor example, iomgr01, iomgr02, qns01, qns02, etc.)
5. Restart the QNS application and enter the command **restartall.sh**