Qualcomm

Qualcomm Technologies, Inc.

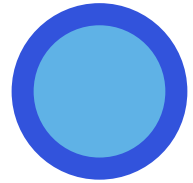
CDSPKBA QualcommProfiler for HTP Usage

80-xxxxx-x Rev. AA

Revision History

Revision	Date	Description
AA	April 2026	

Contents



Section 1

Overview and Installation

Qualcomm Profiler Overview

- The Qualcomm Profiler tool is a **system-wide performance profiling tool** designed to **visualize system performance and identify optimization and application scaling improvement opportunities across Qualcomm SoC CPUs, GPUs, DSPs, and other IP blocks**. Qualcomm Profiler includes **a GUI Application**, Command Line Interface (**CLI**) and **API** - Lightweight C-based API that provides an interface for developers to profile and optimize their applications on Qualcomm platforms.
  - User can enable profiling either from Host to get live profiling metrics or directly on device. Supported for below Device and Host combinations:
    - Device: Android platforms, QNX, Windows on Arm (WoS), K2L
    - Host: Windows 10, Ubuntu 20.04.
  - Supported chipsets
    - **Android:** SM8850, SM8750, SM8650, SXR2230P
- **Key Features**
  - Concurrently profiles different subsystems, such as, GPU, CPU, and NSP.
  - Dynamically discovers profiling capabilities and profiling metrics.
  - Stream profiling data based on configured metrics and capabilities.
  - Configurable profile duration and sampling rate.
  - Qtimer timestamp . It helps visualize performance metrics across subsystems.
  - Sysmon Binary file collection and analysis from Profiler UI
  - Collects profiling metrics in Perfetto trace event format.
  - NSP roofline analysis support
  - QUTS based automation support.
  - More can check from ReleaseNotes.txt

Sysmon Migration

- SysmonApp is an older application which interacts with DSP subsystem via FastRPC and provides various users functionalities such as profiling, set/remove bus clock, get software thread info, and other functionalities.
- **We have migrated the Sysmon App and Sysmon Parser functionality to Qualcomm profiler. Any new functionality will be added via Qualcomm Profiler .**

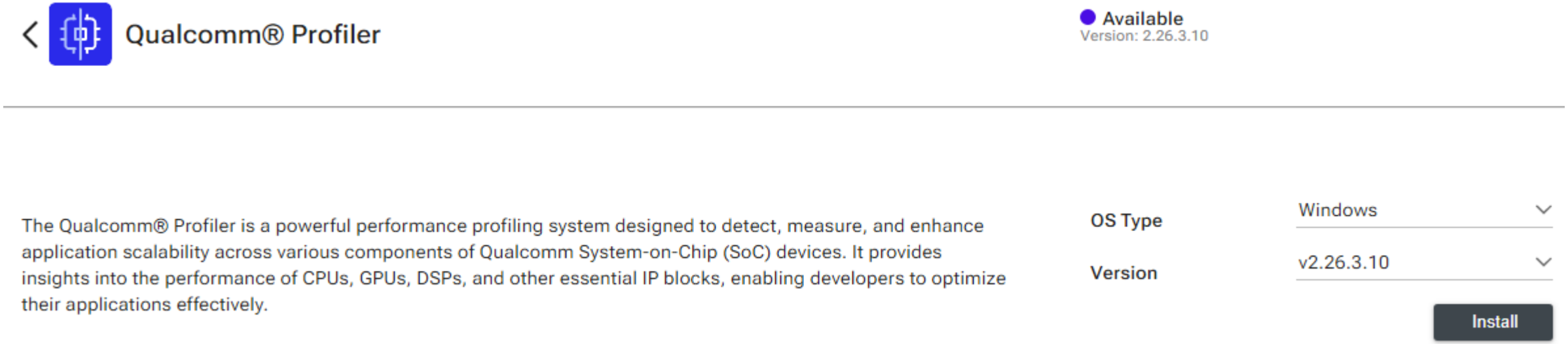
Feature	SysmonApp	Qualcomm profiler
profiler	Y	Y
TLP profiling	Y	Y
getLoad	Y	Y (Live metrics)
getState	Y	Y
getInfo	Y	Y
pInfo	Y	Y
tInfo	Y	Y
getPowerStats	Y	Y
clocks	Y	Y (profilerUtilityApp)
benchmark	Y	Y (profilerUtilityApp)
etmTrace	Y	Y (profilerUtilityApp)
DCVS	Y	Y (profilerUtilityApp)
bin parser	Y	Y (qpbin_parser)
Multiple DSP profiling	N	Y
Multiple capabilities profiling	N	Y
System capabilities (CPU, GPU, thermal, power profiling)	N	Y
On-Host CLI profiling	N	Y
API support	N	Y
On-Host GUI application with charts, graph	N	Y
Roofline analysis	N	Y
NSP Stats profiling with Default profiling	N	Y
Live metric support	N	Y
Perfetto trace format support	N	Y(Not supported for stats)

External Dependencies

- **ADB(for android mobile device)**
  - Download ADB command line tools from Android Developer Platform-Tools.
- **ADB port forwarding (optional)**
  - For Android devices, port forwarding can be used to stream data to the host over localhost.
  - To enable ADB port forwarding, run **adb forward tcp:62472 tcp:62472 on the host**.
  - If this command is successful, you can start profiling the server on 127.0.0.1:62472

Install from QPM

- QPM: Download SnapdragonProfilerSetup_External.exe and install at C:\Program Files\Qualcomm\Snapdragon Profiler



  - The Qualcomm Profiler GUI and command line interface (CLI) are installed at location:
    - Windows X64: C:\Program Files (x86)\Qualcomm\QualcommProfiler

Installation by qpm-cli Command

- Qualcomm Profiler is available in Qualcomm Package Manager
- Please install Qualcomm Package Manager 3 and install Qualcomm Profiler using below commands.

Commands	Windows / Linux / Windows-ARM64
License Activation	qpm-cli --license-activate Qualcomm_Profiler
Installation	qpm-cli --install Qualcomm_Profiler
Uninstallation	qpm-cli --uninstall Qualcomm_Profiler

Device and Host Setup

- Device CLI steup
  - Assuming you have legacy LA device(SM8750 and before), please just run **Installer "**C:\Program Files (x86)\Qualcomm\Shared\QualcommProfiler\API\target-la**InstallerLA.exe"**

- Host CLI setup
  - To set up the Qualcomm Profiler CLI for host, clients(**Windows (X86) host**) must communicate to the server running on the device over IP. Profiling data can only be streamed from Android and QNX devices to a Windows host.
  - Qualcomm profiling **Service configure:**
    - export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/vendor/qprof/libs/ //important
    - export QMONITOR_BACKEND_LIB_PATH=/vendor/qprof/backends
    - export QMONITOR_FRONTEND_LIB_PATH=/vendor/qprof/libs/
    - qprof --configure --server-ip 127.0.0.1 --port 62472
    - qprof --stop-server
    - qprof --start-server //Don't close shell window
  - Qualcomm profiling **client configure** with the IP/port where the Qualcomm profiling server is running:
    - C:\Program Files (x86)\Qualcomm\QualcommProfiler\CLI\host-windows-64\bins>qprof --configure --server-ip 127.0.0.1 --port 62472
      - *Server IP : 127.0.0.1 and port : 62472 have been configured successfully*
    - C:\Program Files (x86)\Qualcomm\QualcommProfiler\CLI\host-windows-64\bins> qprof --capabilities

Reference

- Online website
  - https://docs.qualcomm.com/doc/80-54323-2/topic/ (Latest)
- Local host location (windows)
  - C:\Program Files (x86)\Qualcomm\Shared\QualcommProfiler\API\documents
    - 80-54323-2_AL_Qualcomm_Profiler_User_Guide.pdf

Section 2

Basic usage

Command Line Interface(CLI)

- **All the device tools/configuration files are part of LA software release; There is NO NEED to install Qprofiler from QPM3 for LA OEM working on 8850/8975;**
  - On device CLI (profilerUtilityApp, qprof)-> these tools are already part of KNP vendor image, the json files will be part of PreCS5 or CS release, THIS IS THE MOST importance to promote to OEM once they have CS release.

- Android device tool(qprof/profilerUtilityApp/qpbin_parser) location: C:\Program Files (x86)\Qualcomm\Shared\QualcommProfiler\API\target-la\aarch64\bins
- Sample-configs under C:\Program Files (x86)\Qualcomm\Shared\QualcommProfiler\API\sample-configs
- Windows CLI location: C:\Program Files (x86)\Qualcomm\QualcommProfiler\CLI\host-windows-64\bins\qprof.exe
- Logs and results location:
  - **Host** profiling **results** at C:\ProgramData\Qualcomm\QualcommProfiler\profilingresults\, **log** location “C:\ProgramData\Qualcomm\QualcommProfiler\logs”
  - **Device** profiling **results** at /data/shared/QualcommProfiler/profilingresults/, **log** by “adb logcat | grep QPROF”

- OEM could refer to https://docs.qualcomm.com/doc/80-54323-2/topic/command-line-interface.html for the command.
- **More new features are always on the way.**

Host/Device Supported Commands List

- Host（windows） supports the following commands:

```
C:\Program Files (x86)\Qualcomm\QualcommProfiler\CLI\host-windows-64\bins>qprof -h
Supported Options::
-h    [--help]             Display this Help Message
-v    [--version]          Display the version number
-lapp [--launch-app]       Launch Application with Specified Arguments
-ca   [--capabilities]     Get available Profilers for the Device
-pr   [--profile]          Start Profiling with Specified Arguments
-genc [--generate-config] Generate the configuration for the capabilities
-c    [--configure]        Configure the profiler CLI
-gsi  [--get-server-info] Display information about configured server ip and port
-cmds [--commands]         Get available commands for the Device
-exec [--execute-command] Execute command on the Device
-mi [--metrics-info]       Show available metrics information for the device, based
 on supported capabilities
```

Host/Device Supported Commands List

- Device（android）supports the following commands:

```
sun:/vendor/bin # qprof -h
Supported Options::
-h    [--help]            Display this Help Message
-v    [--version]         Display the version number
-lapp [--launch-app]      Launch Application with Specified Arguments
-ca   [--capabilities]    Get available Profilers for the Device
-pr   [--profile]         Start Profiling with Specified Arguments
-genc [--generate-config] Generate the configuration for the capabilities
-sts  [--start-server]    Start Profiling Server on the Device
-sos  [--stop-server]     Stop Profiling Server on the Device
-c    [--configure]       Configure the profiler CLI
-gsi  [--get-server-info] Display information about configured server ip and port
-cmds [--commands]        Get available commands for the Device
-exec [--execute-command] Execute command on the Device
-sm [--sysmon]            Sysmon Profiling with Specified Arguments
-mi [--metrics-info]      Show available metrics information for the device, based on supported capabilities
```

- Device CLI starting support Sysmon like short command option **“-sm [--sysmon]”** from build 2.26.2.x and there will be no config file changes required. Will integrated same to build as well.
- For Device with old version Qprofiler, pls install latest Qualcomm Profile from QPM3.Assuming you have SM8850 LA device, please just run C:\Program Files (x86)\Qualcomm\Shared\QualcommProfiler\API\target-la**InstallerLA.exe**, which will update the dependency files to LA target.

Some Command Options

- --launch-app: This command launches the application on the device. It is only available on the device client.
- --capabilities- list [-clist] (required) :List of profilers to enable
- --profile <#_bookmark21> [-pr]:Starts profiling with specified arguments
- --launch-app [-lapp]:Launches application with specified arguments
- --configure [-cn] args:Configures profiler CLI
- --generate-config : This command generates a configuration for a user entered capability.

Some Command Options

- **qprof –sm :**QProfiler Sysmon like short command support

sysMonApp getinfo --q6 cdsp	qprof -sm getinfo --q6 npu0
sysMonApp getstate --q6 cdsp	qprof -sm getstate --q6 npu0
sysMonApp getstate --getvotes 1 --q6 cdsp	qprof -sm getstate --getvotes 1 --q6 npu0
sysMonApp pinfo --q6 cdsp	qprof -sm pinfo --q6 npu0
sysMonApp pinfo --maxt 1 --q6 cdsp	qprof -sm pinfo --maxt 1 --q6 npu0
sysMonApp tinfo --q6 cdsp	qprof -sm tinfo --q6 npu0
sysMonApp profiler --q6 cdsp--duration 5	qprof -sm profiler --q6 npu0 --duration 5
sysMonApp tlp --q6 cdsp--samplingPeriod 100 --duration 10 --enableRoot 0	qprof -sm tlp --q6 npu0 --samplingPeriod 100 --duration 10 --enableRoot 0
sysMonApp tlp --q6 cdsp --samplingPeriod 100 --duration 10 --enableRoot 1 --profile 1	qprof -sm tlp --q6 npu0 --samplingPeriod 100 --duration 10 --enableRoot 1 --profile 1

Command Examples

- **“qprof --capabilities”** lists all subsystem and IP blocks available to profile.
  - C:\Program Files (x86)\Qualcomm\QualcommProfiler\CLI\host-windows-64\bins>qprof --capabilities

```
C:\Program Files (x86)\Qualcomm\QualcommProfiler\CLI\host-windows-64\bins>qprof --capabilities

Friendly Name                                Capability              Streaming Rate(s)        Sampling Rate(s)             Available Metric
(s)
-------------------------------------------------------------------------------------------------------------------------------------------
-------------------------------------
NPU1-Stats                           profiler:nsp1-dsp-stats         1000-2000                1000-2000                    5888-5890 5893-59
14
NPU0-Stats                            profiler:nsp-dsp-stats         1000-2000                1000-2000                    5888-5890 5893-59
14
ADSP-Stats                           profiler:adsp-dsp-stats         1000-2000                1000-2000                    5888-5897 5900 59
03-5910 5912-5914
NPU1                                profiler:nsp1-dsp-metrics        200-1000                 1-10                         4096-4184 4187-41
88 4190-4192 4195 4198-4205 4236-4241 4244-4267 4352 4356
                                                                                                                          4358 4360-4362 43
66-4374 4377-4385 4395-4396 4480-4524
NPU0                                 profiler:nsp-dsp-metrics        200-1000                 1-10                         4096-4184 4187-41
88 4190-4192 4195 4198-4205 4236-4241 4244-4267 4352 4356
                                                                                                                          4358 4360-4362 43
66-4374 4377-4385 4395-4396 4480-4524
ADSP                                profiler:adsp-dsp-metrics        200-1000                 1-10                         4096-4205 4244-42
65
THREAD                    profiler:apps-proc-thread-profiling        200-400                  100-200                      4660
Thermal                    profiler:apps-proc-thermal-metrics        200-1000                 10-160                       6464-6465
Memory                         profiler:apps-proc-mem-metrics        200-1000                 50-200                       4639-4641 4643-46
45 4648-4649
Process                    profiler:apps-proc-process-metrics        200-1000                 50-200                       4642
IO                              profiler:apps-proc-io-metrics        1000                     200-1000                     4646-4647
GPU-Processor              profiler:proc-gpu-specific-metrics        200-1000                 50-70                        4864-4868
DDR                            profiler:apps-proc-ddr-metrics        200-500                  10                           4661-4663
Network                        profiler:apps-proc-net-metrics        200-1000                 50-200                       4656-4659 4678-46
82
CPU-PMU                    profiler:apps-proc-cpu-pmu-metrics        200-1000                 20-200                       0-33133
Process Memory         profiler:apps-proc-process-mem-metrics        200-1000                 50-200                       4683-4686
Battery                    profiler:apps-proc-battery-metrics        200-1000                 50-200                       4703-4708
CPU                            profiler:apps-proc-cpu-metrics        200-1000                 10-200                       4609-4616 4618-46
24 4696 4720 4731-4735 6414 6424-6426 6437-6444 6455-6462
```

Command Examples

- “**qprof --profile**” enables streaming profiling data to the client
  - C:\Program Files (x86)\Qualcomm\QualcommProfiler\CLI\host-windows-64\bins>qprof --profile --profile-type async --file-format json --capabilities-list profiler:apps-proc-ddr-metrics profiler:nsp-dsp-metrics --streaming-rate 500 --result-format verbose --live --metric-id-list 4661

```
C:\Program Files (x86)\Qualcomm\QualcommProfiler\CLI\host-windows-64\bins>qprof --profile --profile-type async --file-format json --capabilities-list
profiler:apps-proc-ddr-metrics profiler:nsp-dsp-metrics  --streaming-rate 500 --result-format verbose --live --metric-id-list 4661

Using default Configuration for capability profiler:apps-proc-ddr-metrics
Provided sampling rate 0 ms is not supported for profiler:apps-proc-ddr-metrics Using nearest sampling rate of 10 ms.
Using default Configuration for capability profiler:nsp-dsp-metrics
Provided sampling rate 0 ms is not supported for profiler:nsp-dsp-metrics Using nearest sampling rate of 10 ms.
Timestamp:13:22:07.283                          Metric ID:4661                                  NOC DDR APPS0 Bandwidth:0.000 MBps

Timestamp:13:22:07.294                          Metric ID:4661                                  NOC DDR APPS0 Bandwidth:880.468 MBps

Timestamp:13:22:07.305                          Metric ID:4661                                  NOC DDR APPS0 Bandwidth:1507.552 MBps

Timestamp:13:22:07.316                          Metric ID:4661                                  NOC DDR APPS0 Bandwidth:1324.705 MBps

Timestamp:13:22:07.327                          Metric ID:4661                                  NOC DDR APPS0 Bandwidth:19.722 MBps

Timestamp:13:22:07.338                          Metric ID:4661                                  NOC DDR APPS0 Bandwidth:15.024 MBps

Timestamp:13:22:07.349                          Metric ID:4661                                  NOC DDR APPS0 Bandwidth:238.687 MBps

Timestamp:13:22:07.360                          Metric ID:4661                                  NOC DDR APPS0 Bandwidth:36.957 MBps

Timestamp:13:22:07.371                          Metric ID:4661                                  NOC DDR APPS0 Bandwidth:18.421 MBps
```

Command Examples

- “qprof -sm getstate --getvotes 1 --q6 npu0”

```
sun:/ # qprof -sm getstate --getvotes 1 --q6 npu0

Using default Configuration for capability profiler:nsp-dsp-stats
Timestamp:15:05:34.191                          Metric ID:5888
DSP Core Clock
DSPCoreClock:1708.800 MHz

Timestamp:15:05:34.191                          Metric ID:5889
Bandwidth Vote
SNOC:0.000 MHz                                  MEMNOC:28.125 MHz

Timestamp:15:05:34.191                          Metric ID:5890
Root PD Heap
Total:4096.000 KB                               Available:1433.027 KB                   Maximum:1318.453 KB

Timestamp:15:05:34.191                          Metric ID:5893
Measured Bus Clock
SNOCClock:0.000 MHz                             MEMNOCClock:403.057 MHz                 BIMCClock:547.345 MHz
CPUL3Clock:0.000 MHz                            HmxClock:249.600 MHz                    Q6CengClock:444.166 MHz
HmxPower:0                                      VotedLpmState:0                         SleepLatency:65535

Timestamp:15:05:34.191                          Metric ID:5894
DCVS Clients
AggregatedBusMin:DISABLE                        AggregatedBusMax:DISABLE                AggregatedBusTarget:DISABLE
AggregatedCoreMin:DISABLE                       AggregatedCoreMax:DISABLE               AggregatedCoreTarget:DISABLE
CoreClockVote:0.000 MHz                         BusClockVote:0.000 MHz                  AggregatedVotedLpmState:0
AggregatedSleepLatency:65535 us                 AggregatedHmxClock:249.600 MHz          AggregatedHmxPower:0
AggregatedCengClock:0.000 MHz                   AggregatedSleepLatencyOverride:0 us     AggregatedFinalSleepLatencyVote:65535 us
DcvsClients:4
-------------------------------------------------------------------------------------------------------------------------------
DCVS Clients: 1
```

Profiler Utility App

profilerUtilityApp is an on-device application that provides the following services:

- Benchmark service
  - Various standard benchmarking algorithms can be run on the NSP to benchmark processor performance. Options can be set for each algorithm, such as defining the height and width of the images on which the benchmarking algorithm's calculations will occur. You can get a list of the supported algorithms and parameters using the following command:
    - *profilerUtilityApp --dsp -benchmark –help*
  - Examples: *profilerUtilityApp --dsp --benchmark -f conv3x3 -w 3840 -h 2160 -p 10 -u 0 -L 1 -l 10000 -n 4*

```
130|sun:/ # profilerUtilityApp --dsp --benchmark -f conv3x3 -w 3840 -h 2160 -p 10 -u 0 -L 1 -l 10000 -n 4
sysmon_init.c : sysmon_init : 356 : INFO : Considering the DSP has old architecture
No q6 Processor provided, hence setting to default : 0
sun

Unsupported q6Processor provided for benchmark, setting to default NSP
q6Processor : 3
Version:2
setting clocks to power level 10
src - allocated 8388608
dst - allocated 8388608
ref - allocated 8388608 (via malloc)
calling conv3x3 on a 3840x2160 image...
Malloc success for LLCC and L3 invalidation, total = 57005
run time of conv3x3: 4848038 microseconds for 1 RPC iterations, each with 10000 iterations inside the DSP.
Average per iteration DSP-measured time : 484 uSec, FastRPC overhead : 3746 uSec, apparent clock rate -1702020314 MHz, HVX units: 4
return value from conv3x3: 0
- success
```

Profiler Utility App

- Clocks service
  - The clocks feature enables setting, limiting, and removing the Q6 clocks of a Q6 processor. The clock value may not be set to the exact value; rather it is set to the nearest supported clock value.
  - Set : Sets the Q6 clock to a desired frequency for the a particular Q6 processor.
    - *profilerUtilityApp --dsp --clocks set --cengClock 700*

```
sun:/ # profilerUtilityApp --dsp --clocks set --cengClock 700 --Q6 cdsp
sysmon_init.c : sysmon_init : 356 : INFO : Considering the DSP has old architecture
Calling q6Processor : nsp set clocks function with following parameters:
Q6->CENG bus clock : 700 MHz
Successfully set the required clock configurations, Call the remove API once done...
```

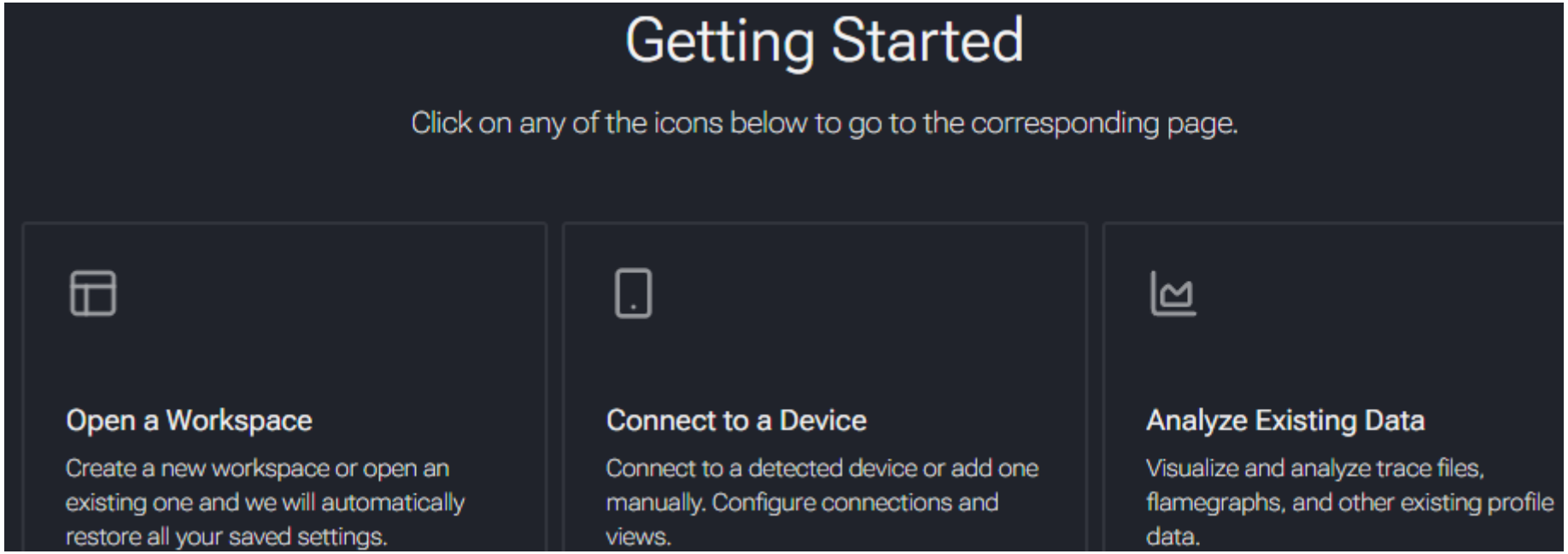
  - Limit : Limits the maximum clock frequency of the Q6 processor at which it can work for the performance
  - Remove : Resets all the clock settings to the default values.
    - *profilerUtilityApp --dsp --clocks remove*

Section 2.3

Qualcomm profiler GUI Application

Launch and Connect to a device

- **Launch the Qualcomm Profiler executable as a desktop application from the install location:**
  - C:\Program Files (x86)\Qualcomm\QualcommProfiler\GUI\qcprofiler.exe(windows) -> low priority to promote to OEM
- **Connect to a device**
  - There are currently two supported types of device connections for profiling:
    - gRPC: Currently Windows X64 and Linux systems support gRPC connection.
    - Localhost: Currently only Windows Arm64 systems support Localhost connections

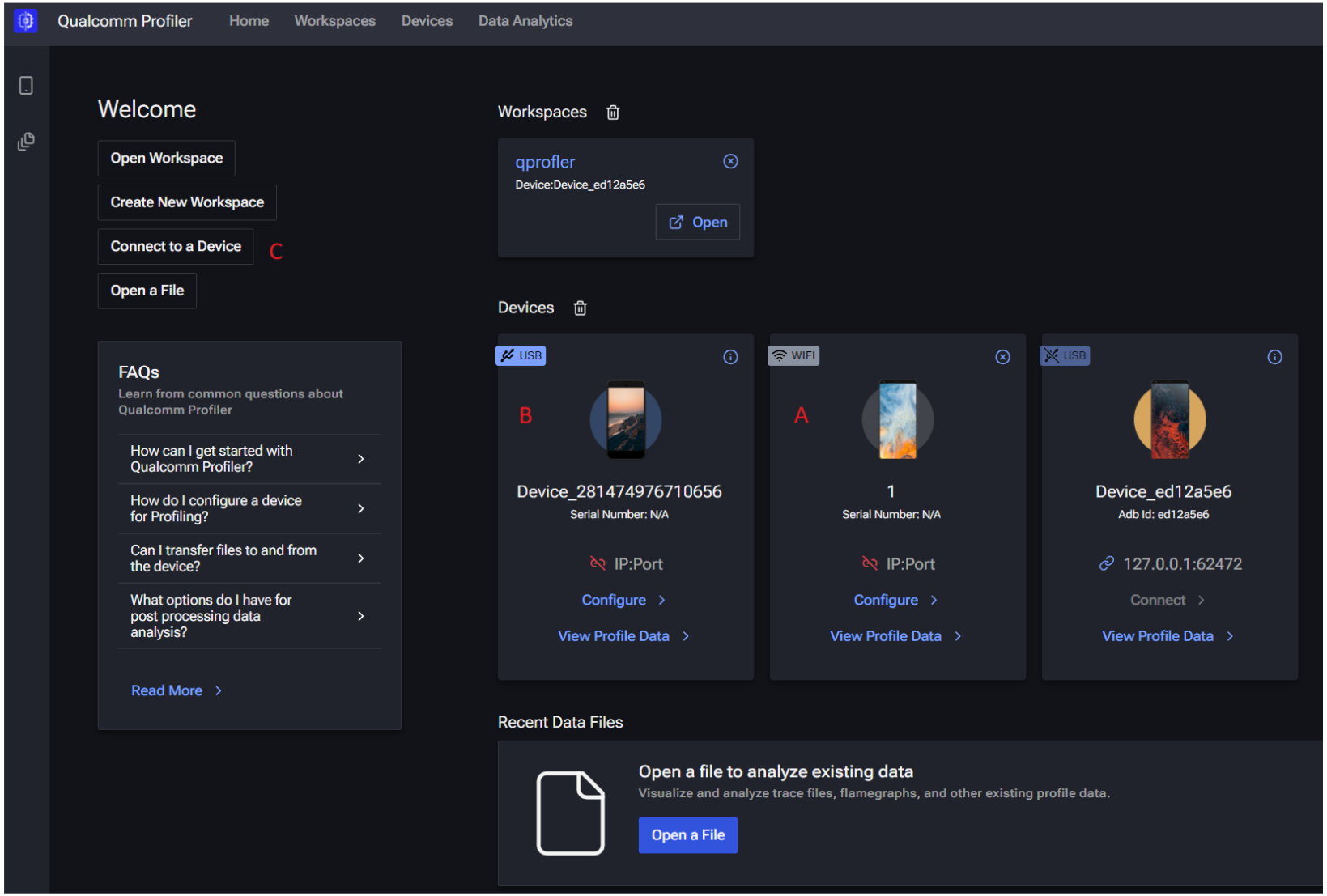


GRPC connection

- To set up a GRPC connection with below steps:
  - 0. Prepare
    - On device
      - export QMONITOR_BACKEND_LIB_PATH=/vendor/qprof/backends
      - export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/vendor/qprof/libs/
      - export QMONITOR_FRONTEND_LIB_PATH=/vendor/qprof/libs/
      - qprof --stop-server
      - qprof --configure --server-ip 127.0.0.1 --port 62472
      - qprof --start-server
    - On host
      - adb forward tcp:62472 tcp:62472
      - qprof --configure --server-ip 127.0.0.1 --port 62472
      - Command to check if OK: “qprof –capabilities”

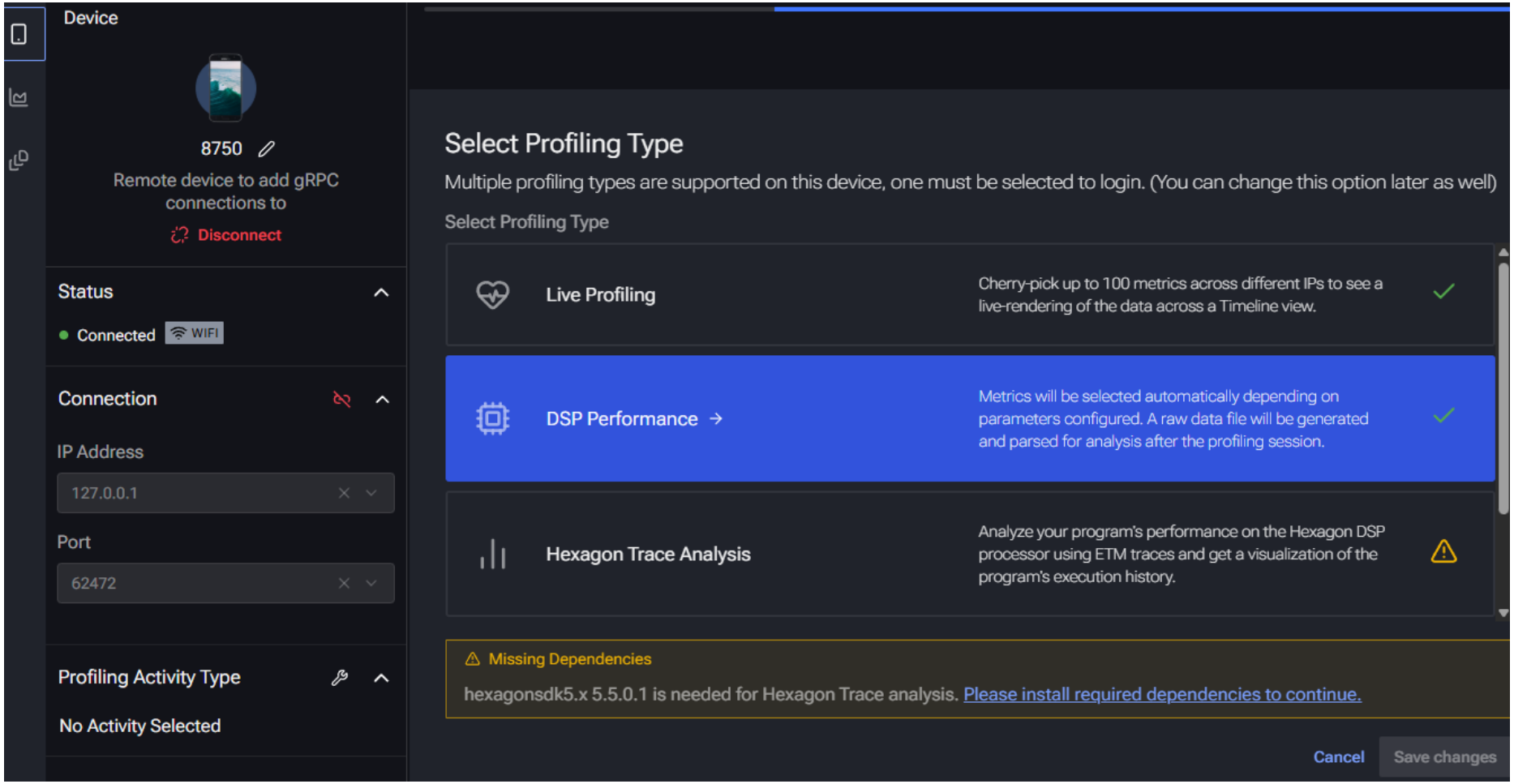
GRPC connection(cont.)

- To set up a GRPC connection with below steps:
  - 1. Select a device
    - The application start page displays cards for previously connected remote devices (see A in the image) and detected devices (see B in the image). Click **Connect** in the card for the device you want to use. Android LA devices are automatically detected when connected via USB to the host machine.
    - To connect to a new remote device, click Connect to a Device (see C in the image). For the new device, enter an alias name and select the platform type. Click **Connect**.



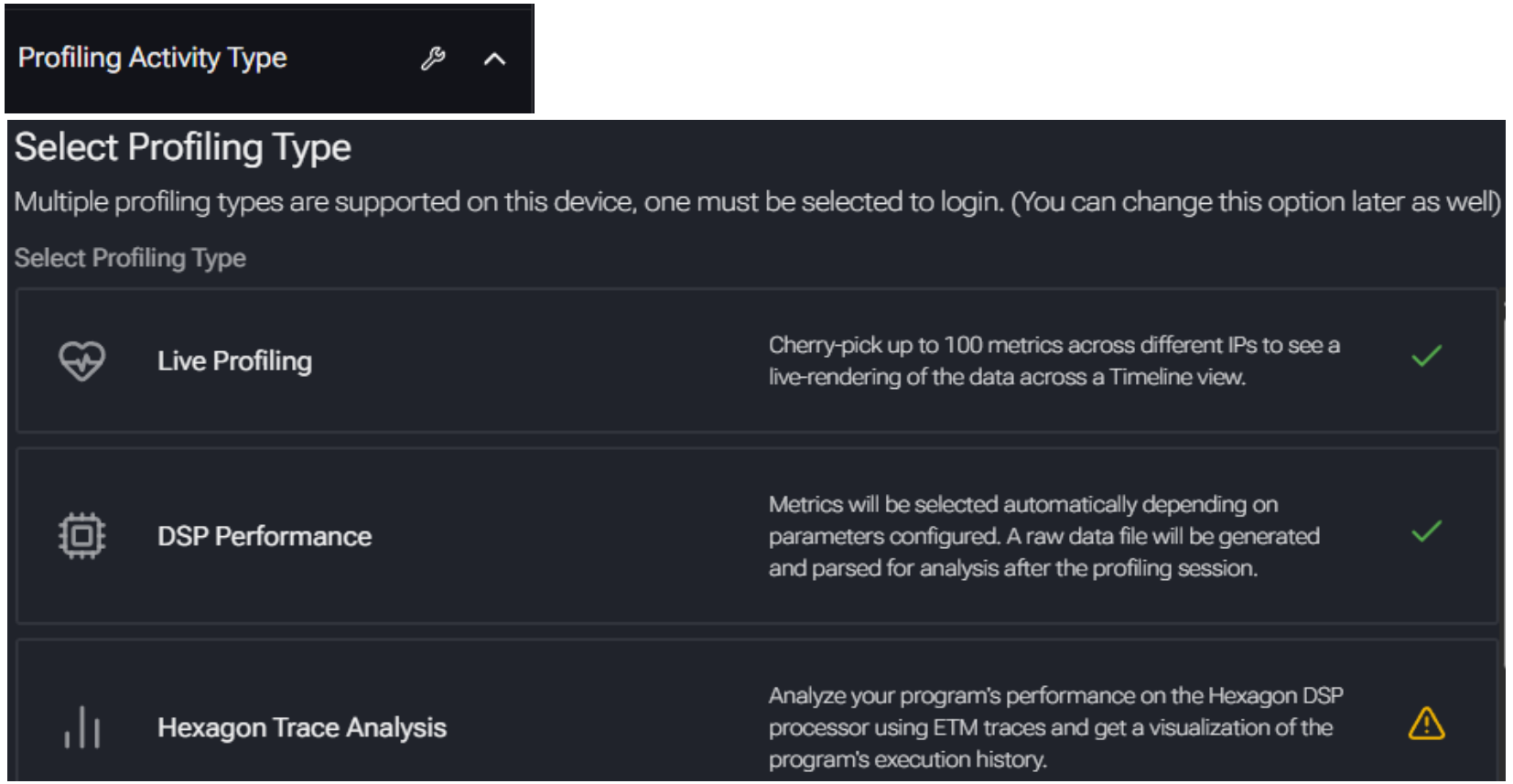
GRPC connection(cont.)

- 2. Enter the IP Address(127.0.0.1) and port number(62472) of the connection.
- 3. Click Connect. When the gRPC connection is successfully opened, the device status show green “Connected”.



Setup Profiling Type

- Upon establishing a connection, a prompt will appear to select the profiling type.
- Once a profiling type is selected, the application will open the respective page for that profiling type's configuration. Then click start and stop to get proper performance statistics.
- Also, can configure "Profiling Activity Type" to a different profiling type.



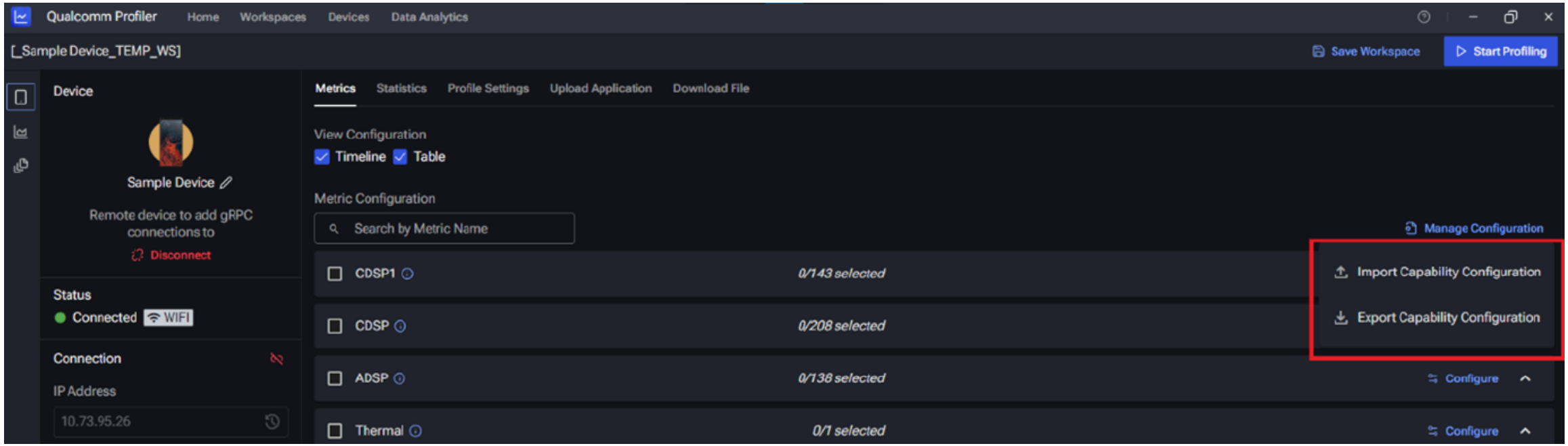
Manage workspaces

- Workspaces help save profiling setup information. A workspace can have one device associated with it and all information about the device will be saved (gRPC IP address and port, adb ID, alias,etc.). All metrics selected, along with any profiling settings, will also be saved to the workspace.
- **Create a workspace**
  - 1. From the Home or Workspaces page, click **Create New Workspace**.
  - 2. Select a device from the recent devices list or add a new one by clicking **New Device** and entering a name. Click **Next**.
  - 3. Enter in an IP address and port if establishing a gRPC connection. Select the desired views. Click **Next**.
  - 4. If necessary, change any profile settings. Click **Next** when complete.
  - 5. Select any file to upload to the target path on the selected device and enter any desired commands to be executed. Click **Save** when satisfied and save the workspace file to a location on the local machine.
- **Open a workspace**
  - From the Home or Workspaces page, click the **Open** button associated with a workspace saved in the Recent panel or click **Open Workspace** to browse the local machine for a workspace。
  - The application will attempt to connect to the saved IP address and port. It will also attempt to restore all the profiling settings, configuration metrics, and their parameters. An error will display if something in the restore process went wrong.

Manage configuration files

Configuration files contain information about selected metrics and their parameter configurations. Users can export and import these configurations.

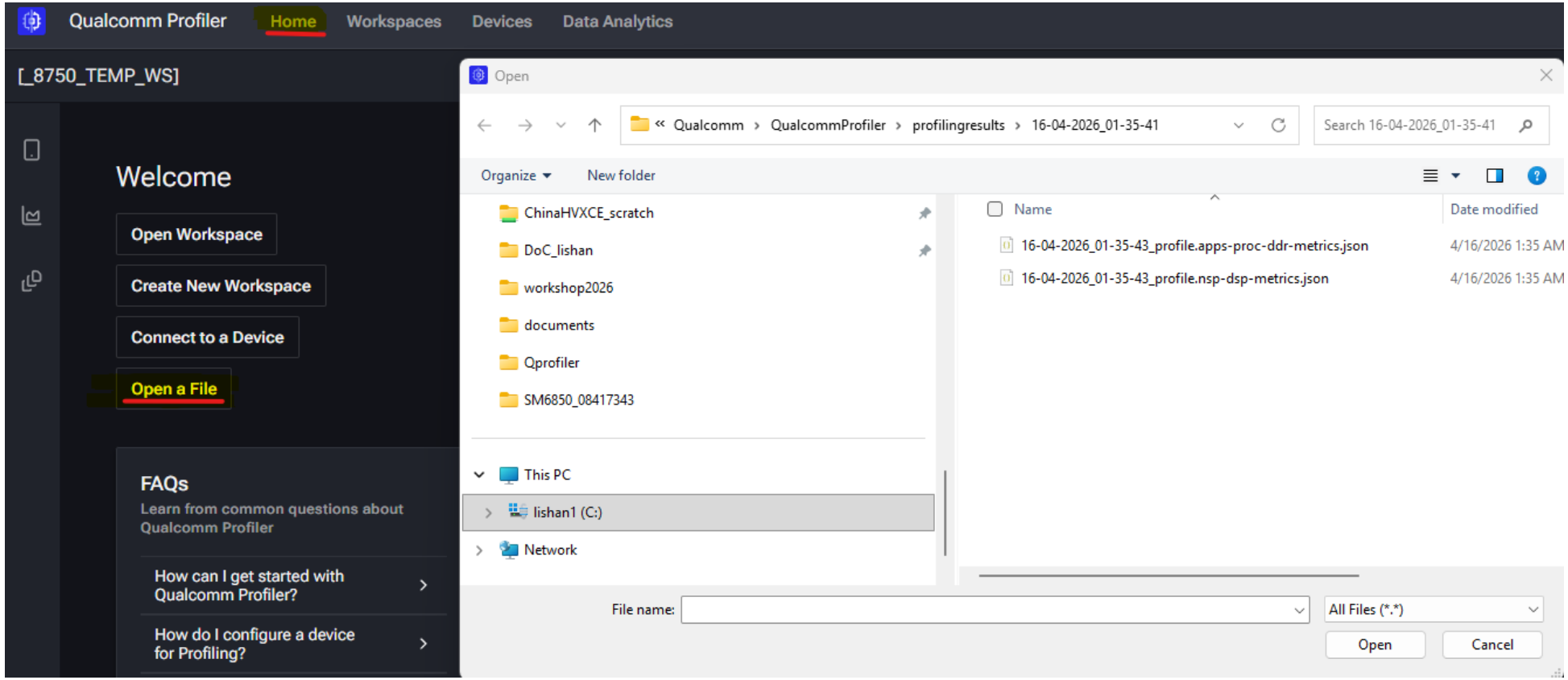
- **Export a configuration file**
  - After connecting to a device, metrics and/or statistics can be selected and further configured by clicking **Configure**. Any selections and configurations can be exported by clicking **Manage Configuration** and then **Export Configuration**.



- **Import a configuration file**
  - After connecting to a device, users can import a configuration file that they have used to configure a similar device. The application will attempt to match all the metrics and settings in the configuration file. Click **Manage Configuration** and then **Import Configuration** to browse your local file system and select a file to import.

Perform postprocessing data analysis

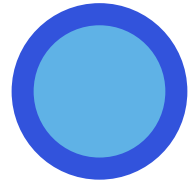
- The Qualcomm Profiler GUI can be used to open various data visualization files. The following file types are supported:
  - Json —— Trace files
  - SVG —— Flame graphs
  - QDB —— Profiled data files
- Open a data visualization file by clicking **Open File** on the Start page or in the Data Analysis tab.



APIs & Sample

The QTI profiler API library provides the interface to collect profiling data on the device

- **Windows (x86)**
  - Header file: C:\PROGRA~2\Qualcomm\Shared\QualcommProfiler\API\include
  - Library (Android): C:\PROGRA~2\Qualcomm\Shared\QualcommProfiler\API\target-la\libs\.so
  - Sample code: C:\PROGRA~2\Qualcomm\Shared\QualcommProfiler\API\sample-code\C++
  - For additional API documentation: C:\PROGRA~2\Qualcomm\Shared\QualcommProfiler\API\documents\api-definition\html_q_profiler_api_8h.html
- **Linux**
  - Header file:/opt/qcom/Shared/QualcommProfiler/API/include/
  - Library (Android):/opt/qcom/Shared/QualcommProfiler/API/target-la/libs/libQualcommProfilerApi.so
  - Sample code:/opt/qcom/Shared/QualcommProfiler/API/sample-code/C++
  - For additional API documentation:/opt/qcom/Shared/QualcommProfiler/API/documents/api-definition/html/_q_profiler_api_8h.html
- **Windows on Snapdragon**
  - Header file: C:\Program Files(Arm)\Qualcomm\Shared\QualcommProfiler\API\include
  - Sample code: C:\Program Files(Arm)\Qualcomm\Shared\QualcommProfiler\API\sample-code
  - For additional API documentation: C:\Program Files(Arm)\Qualcomm\Shared\QualcommProfiler\API\documents\apidefinition\html_q_profiler_api_8h.html



Section 3

CDSP Performance Profiling

CDSP(NSP) Performance Profiling Ways

- Qualcomm Profiler can be used to gather information like the resource usage, load distribution across available hardware threads, and various other profiling metrics useful in measuring performance, debugging performance related issues, and identifying possible optimizations.
- DSP performance profiling can be done in the following ways:
  - Command Line Interface - Lightweight, on-device CLI that can be executed in a device shell to collect various profiling metrics for the device filesystem. Can later be analyzed using the Qualcomm Profiler GUI.
  - Qualcomm Profiler GUI Application - GUI application running on a host PC used to collect various performance metrics from the device.
  - **API** - Lightweight C-based API that provides an interface for developers to profile and optimize their applications on Qualcomm platforms.
- OEM could refer to https://docs.qualcomm.com/doc/80-54323-2/topic/dsp-profiling.html

Command Line Interface

- Qualcomm Profiler CLI can be used to do DSP performance analysis by **“qprof –profile”/ “qprof -sm profiler”/ “qprof -sm tlp”**command. NSP profiling data can be collected in two modes:
  - Live profiling： Shows profiling results on stdout.
  - Binary mode： Save **Binary files.** Can be postprocessed in the Qualcomm Profiler UI and users can configure the profiling data mode using the Generate Config.

- Qprof --profile : Starts profiling with specified arguments.
  - sun:/ # qprof --profile -h
  - It profiles the sync or async command. Sync calls involves command & response, Async calls involves streaming trace data to the client
  - **Example Usage** : qprof --profile --application Path_To_Executable --application-arguments [Arg1,Arg2] --trace-options Config_Json --profile-type async **--capabilities-list** capability1 capability2 --profile-time 5 --streaming-rate 500 --sampling-rate 100 --filter filter1 filter2 --metric-id-list metricId1 metricId3:metricId5 metricId7 --live --result-format verbose/trace/csv/qpbin --analysis roofline-system --summary --roofline-operationType scalar --roofline-operationSubType float --roofline-pass 2 --pmu-event-list pmuEventId1 pmuEventId2 --result-dir-path C:/temp/resultdir

qprof --profile

- qprof --profile <#_bookmark21> [-pr]: Starts profiling with specified arguments.

```
Description:
-app          [--application]           (Optional) : To start some application before starting Profiling, takes path to executable.
-apparg       [--application-arguments] (Optional) : Application arguments.
-tops         [--trace-options]         (Optional) : Configuration to be passed to profiler.
                                                     If not passed default configuration will be used
-ptype        [--profile-type]          (Optional) : <sync/async> Async calls involves streaming trace data to the client.
                                                     Currently, only async mode is supported.
-clist        [--capabilities-list]    (Mandatory) : List of profilers to be enabled . More than 1 profiler can be enabled.
-ptime        [--profile-time]          (Optional) : Duration in seconds for which profilers need to  be enabled.
                                                     By default profiling will be enable for 10 sec.
-strate       [--streaming-rate]        (Optional) : Time interval in milliseconds between 2 profiling packets recieved on the client.
                                                     By default user will recieve packet every 500 ms
-sprate       [--sampling-rate]         (Optional) : Time interval in milliseconds between 2 profiling samples collected from the device.
                                                     By default sampling rate will be same as streaming rate.
-f            [--filter ]               (Optional) : Show the profiling result of specific parameter.
-milist       [--metric-id-list]        (Optional) : Filter results for particular IDs.
-l            [--live]                  (Optional) : Live streaming of data to standard output . By default live will not be enabled.
-resfmt       [--result-format arg]     (Optional) : Capture Profiling Data in verbose or trace event format. Supported values (verbose/trace/csv).
-anal         [--analysis arg]          (Optional) : Option to enable roofline profiling. Supported values (profile/roofline-system/roofline-functiona
l/ops).
-smry         [--summary]               (Optional) : Print summary Min , Max & Avgerage at the end. By default summary will not be enabled.
-pelist       [--pmu-event-list]        (Optional) : collect defined pmu event Ids profiling data.
-rdirp        [--result-dir-path]       (Optional) : Directory path to save profiling result file
-roofoprtype  [--roofline-operationType]           : Option to provide roofline operation type. Supported values (scalar/vector/matrix).
-roofoprstype [--roofline-operationSubType]        : Option to provide roofline datatype. Supported values (fixed/float).
-roofpass     [--roofline-pass]         (Optional) : Perform roofline for the provided pass number. By Default roofline is performed for the supported
 pass numbers.
-tab          [--tabular]               (Optional) : Display metrics in tabular format. By Default verbose format is used.
```

qprof -sm

- Device CLI sysmon-migration
  - **Pls refer more from https://docs.qualcomm.com/doc/80-54323-2/topic/dsp-profiling.html#sysmon-migration**
- **1. profiler:** collect performance statistics of Q6
  - Example: qprof -sm profiler --q6 npu0 --duration 5 --defaultSetEnable 1 --samplingPeriod 100 --dcvsOption 0 --profileFastrpcTimeline 0 --enableRoot 0 --profileLPMLA 0

```
Available Options for adsp/npu0/sdsp...: (If no argument then uses default):
   --q6                      (default selected processor:adsp):
   --defaultSetEnable        (default:1 - default mode) Input 0 for user mode
                                 0 - User mode logging with 4 PMU events per set
                                 1 - Default mode logging
                                 2 - User mode logging with 8 PMU events per set
                                 3 - User mode logging of all available PMU events with 8 PMU events per set
   --profileFastrpcTimeline        Profile the time taken by fastRPC communication
                                   Enabled by default for ADSP processor sysmon profiler
                                   Provide below command to enable fastrpc logging for desired process
                                   echo 0x001f001f00000000> /vendor/lib/rfsa/adsp/Process_name.farf
                                   Process_name can be obtained from adb shell ps -A
                                      0 - Disable
                                      1 - Enable
   --profileLPMLA            Profile Low Power Machine Learning Accelerator block in LPASS (default = 0)
                                 0 - Disable
                                 1 - Enable
   --samplingPeriod          sampling period in ms(default:0 ms - no sampling period override - to be decided by qprof
         Following options are valid only for 'Default mode' logging
   --duration                profiling duration in seconds(default:10s)
   --dcvsOption                  1 - Keep DCVS enabled (if not disabled externally by other clients) for this profiling
                                 0 - disable DCVS for this profiling. Use this option if you want to fix a particular sampling rate.
                             (Default: 0 - Disables DCVS)
   --pmu-event-list          (Optional) collect defined pmu event Ids profiling data.
```

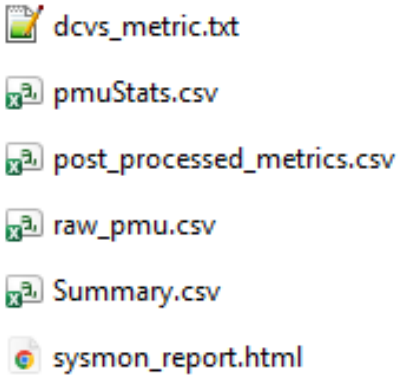
qprof -sm (cont.)

- **2. tlp:** collect thread level performance statistics of Q6
  - Only supported in Binary mode
  - Example: qprof -sm tlp --q6 profiler:nsp-dsp-metrics --samplingPeriod 100 --duration 10 --enableRoot 1 --profile 1

```
Available Options: (If no argument then uses default):
   --q6                    (default selected processor:ADSP):
                               adsp - Selected processor ADSP
                               sdsp - Selected processor Sensors DSP
                               npu0 - Selected processor NPU0
   --samplingPeriod        sampling period in ms(default:50 ms)
   --duration              profiling duration in seconds(default:10s)
   --enableRoot            Profiles UserPD and RootPD threads
                               0 - UserPD threads only (default)
                               1 - User and RootPD threads
   --tName                 Thread names to profile seperated with comma(default: Profile all active threads)
   --profile               Runs Sysmon profile in paralle with TLP(default:0)
                               0 - Sysmon Profile is not executed
                               1 - Sysmon Profile is executed
```

Parsing profile Binary mode files

- **Binary mode:** Parsing bin file
  - Parsing profile .qpbin file step:
    - 1. C:\wordir\doc\mybakfromssd\workshop2026> adb pull /data/shared/QualcommProfiler/profilingresults/24-10-2024_07-02-47/ ./
    - 2. C:\Program Files (x86)\Qualcomm\Shared\QualcommProfiler\API\host-windows-64\bin\qpbin_parser.exe" <input_file> --outdir <output_folder>
      - qpbin_parser.exe C:\wordir\doc\mybakfromssd\workshop2026\24-10-2024_07-02-47\24-10-2024_07-08-30_profile.nsp-dsp-metrics-0.qpbin --outdir C:\wordir\doc\mybakfromssd\workshop2026\24-10-2024_07-08-28\
  - Parsing tlp bin file:
    - C:\Program Files (x86)\Qualcomm\Shared\QualcommProfiler\API\host-windows-64\bin\qpbin_parser.exe" --tlp <input_file> --outdir <output_folder>
- Parsing output:



CLI Profiling and example

- sysmon_report.html

Generated .csv reports

post_processed_metrics.csv -- Detailed PostProcessed metrics
raw_pmu.csv -- Raw PMU events and values
pmuStats.csv -- Accumlated all PMU events and details

Overall Summary

Start time	End time	Total Time in sec	Deep sleep time in sec	Idle time in sec	Active time in sec	MPPS	QDSP6 load in MHz	AXI_RD_BW in MBps	AXI_WR_BW in MBps	QDSP6 Utilization %
685:48:30:76	685:48:30:162	0.086	0.000	0.000	0.083	469.08	1422.67	0.00	0.00	81

Processor summary: Help

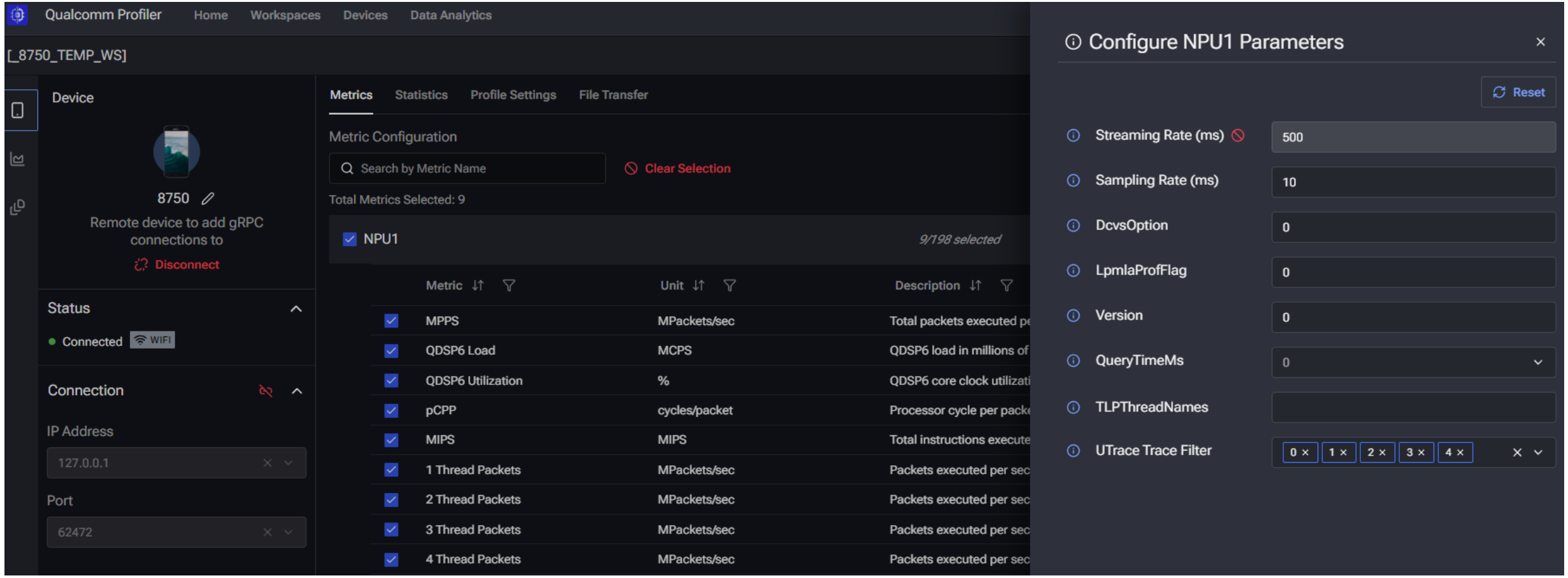
Metric	Unit	Average	Maximum
Core Metrics Help			
MPPS	MPackets/sec	469.08	905.65
pCPP	cycles/Packet	3.03	22.32
QDSP6 load	MCPS	1422.67	1708.89
Bus Metrics Help			
AXI_RD_BW_128B	MBps	210.88	1345.16
AXI_WR_BW_128B	MBps	66.11	1071.64
Clock Metrics Help			
QDSP6 load	MCPS	1422.67	1708.89
Final_DSP_Clock	MHz	1708.80	1708.80
Static_core_clk_vote	MHz	1700.00	1700.00

GUI Profiling and example

- GUI Application supported three types of profiling:
  - Live profiling
  - DSP Performance profiling.
  - Hexagon Trace Analysis

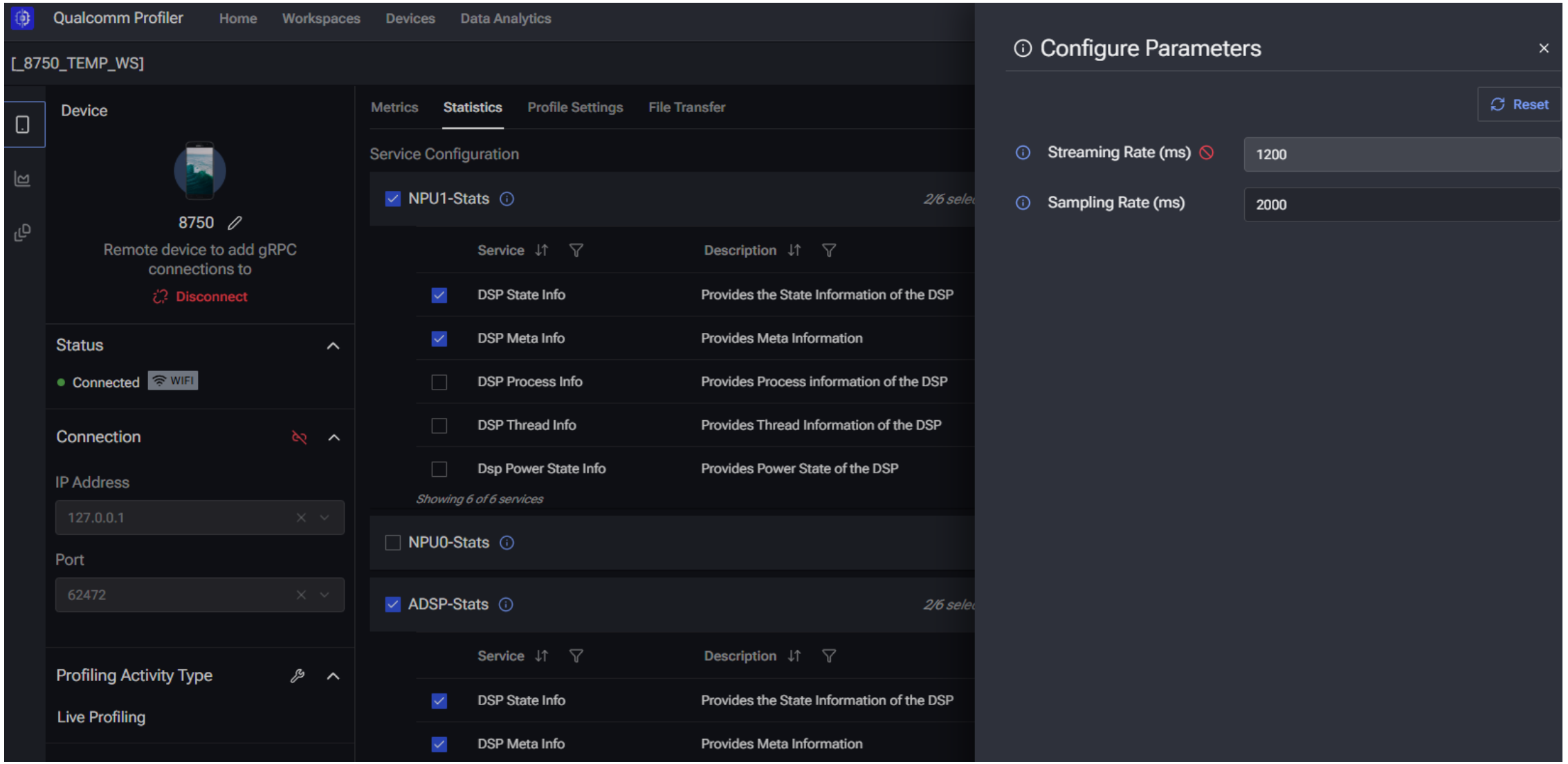
Live profiling

- Select from up to 100 metrics across different IPs to see a live rendering of the data across a timeline and table view. To set up the connection for profiling:
- 1. After the connection is successfully established, the categories that the device supports for profiling are shown along with the metrics associated with each category that can be profiled. Select the metrics and/or statistics to profile and how to view them (timeline or table). A maximum of 10 selections are allowed.



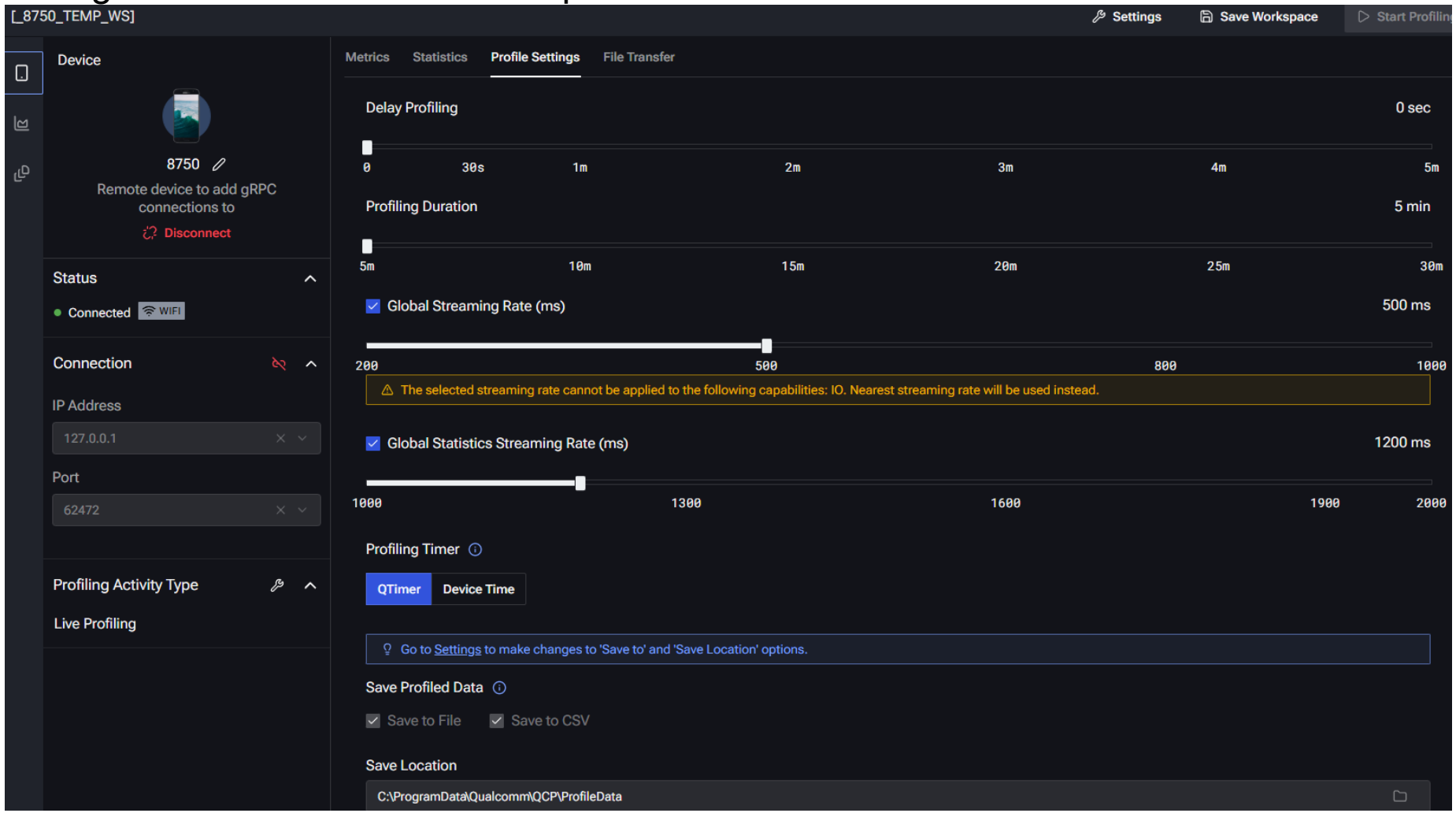
Live profiling(cont.)

- Click the **Statistics** tab to make any statistics selections.



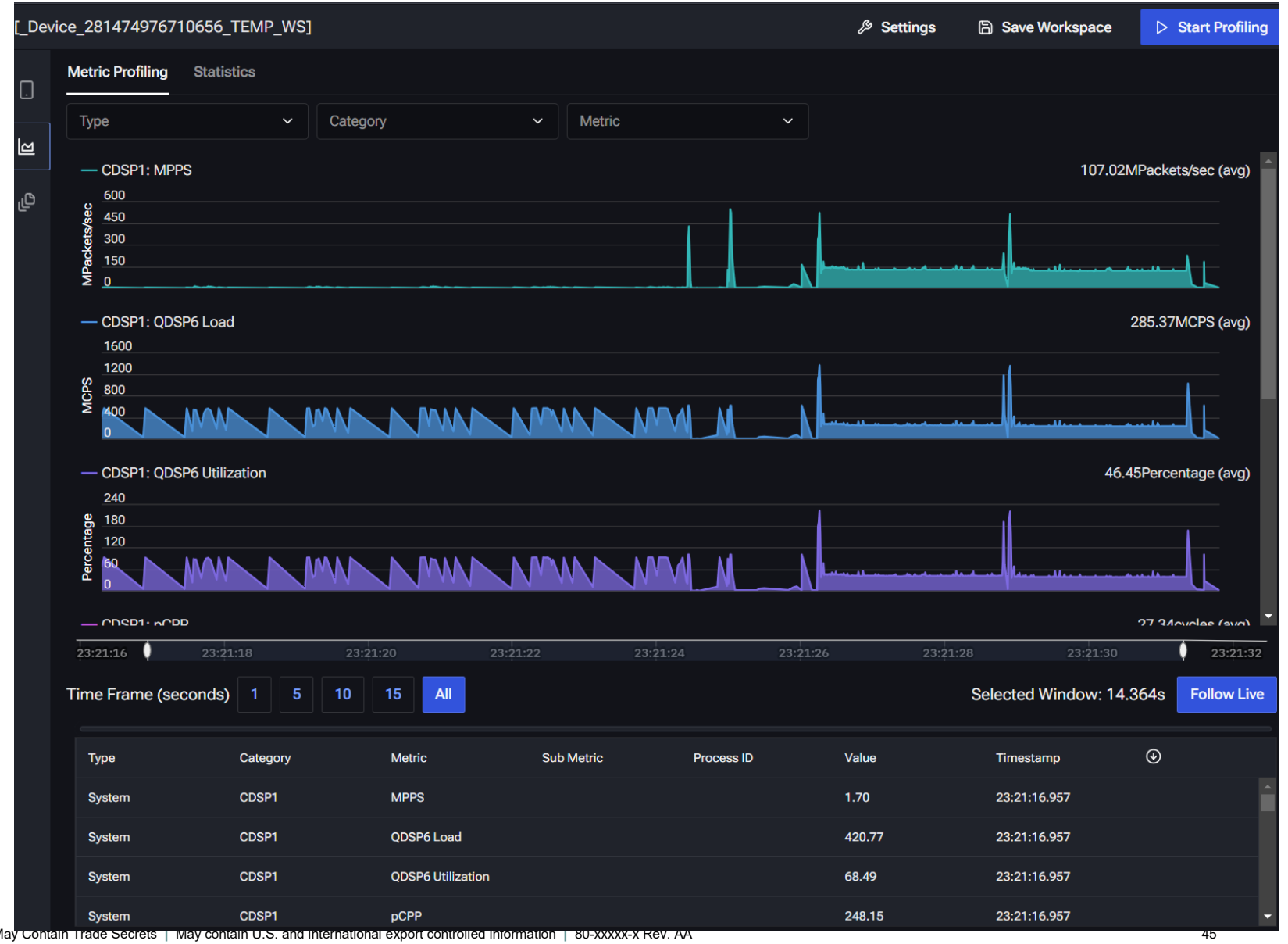
Live profiling(cont.)

- 2.Click the Profile Settings tab. Make selections for delay profiling, profiling duration, streaming rate, and profiling timer. Choose to save the profiled data to a file and/or a CSV.



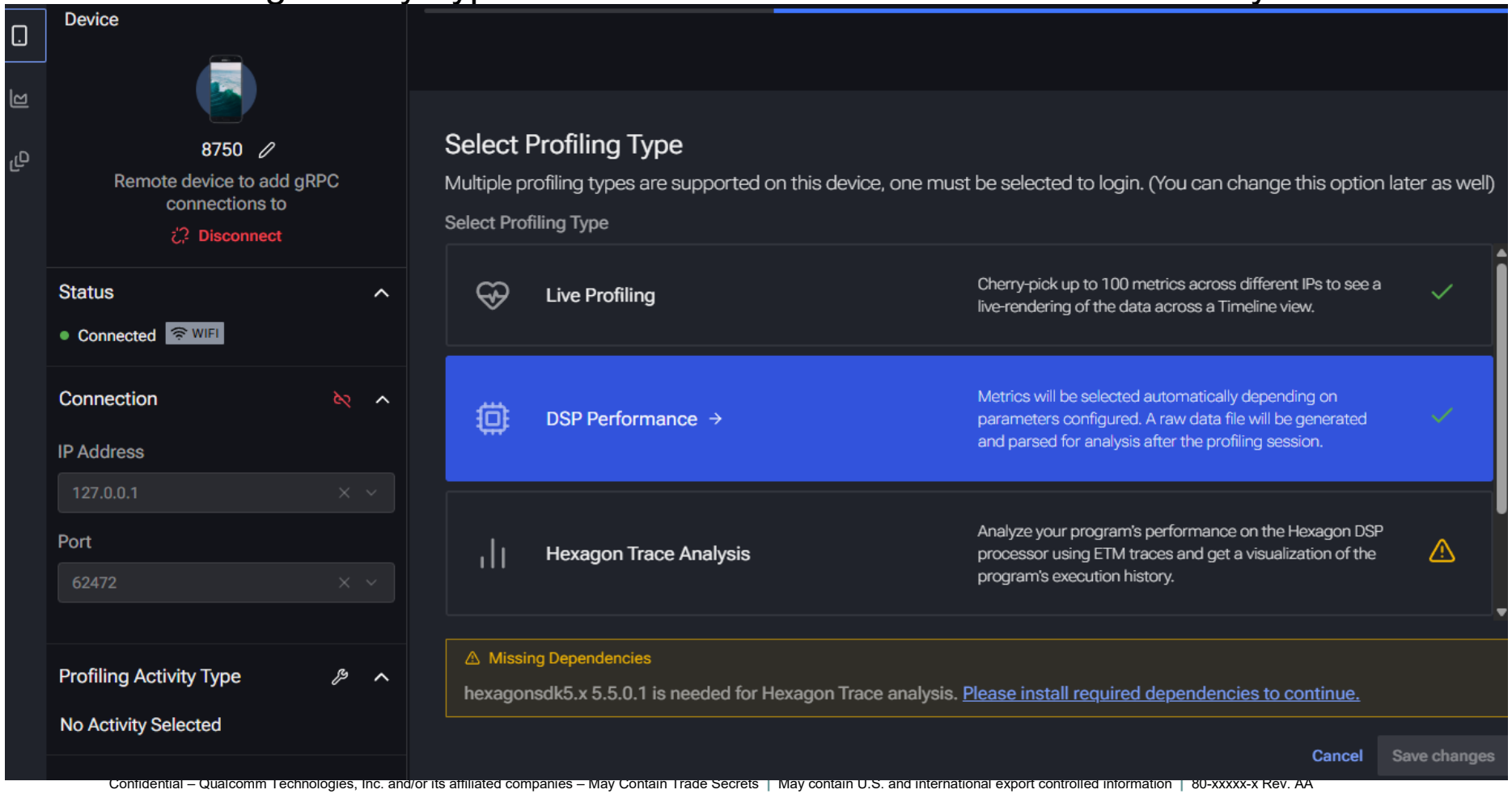
Live profiling(cont.)

- 3. Click Start Profiling to start the profiling session. The Live Profiling View displays once profiling starts.
- 4. Wait until the selected profiling time elapses or click Stop Profiling to end the profiling session at any time.
- 5. View the data collected from the profiling session in the view tab:
  - Timeline View - Use the filters at the top of the profiling session view to filter the charts by metric, category, and type.



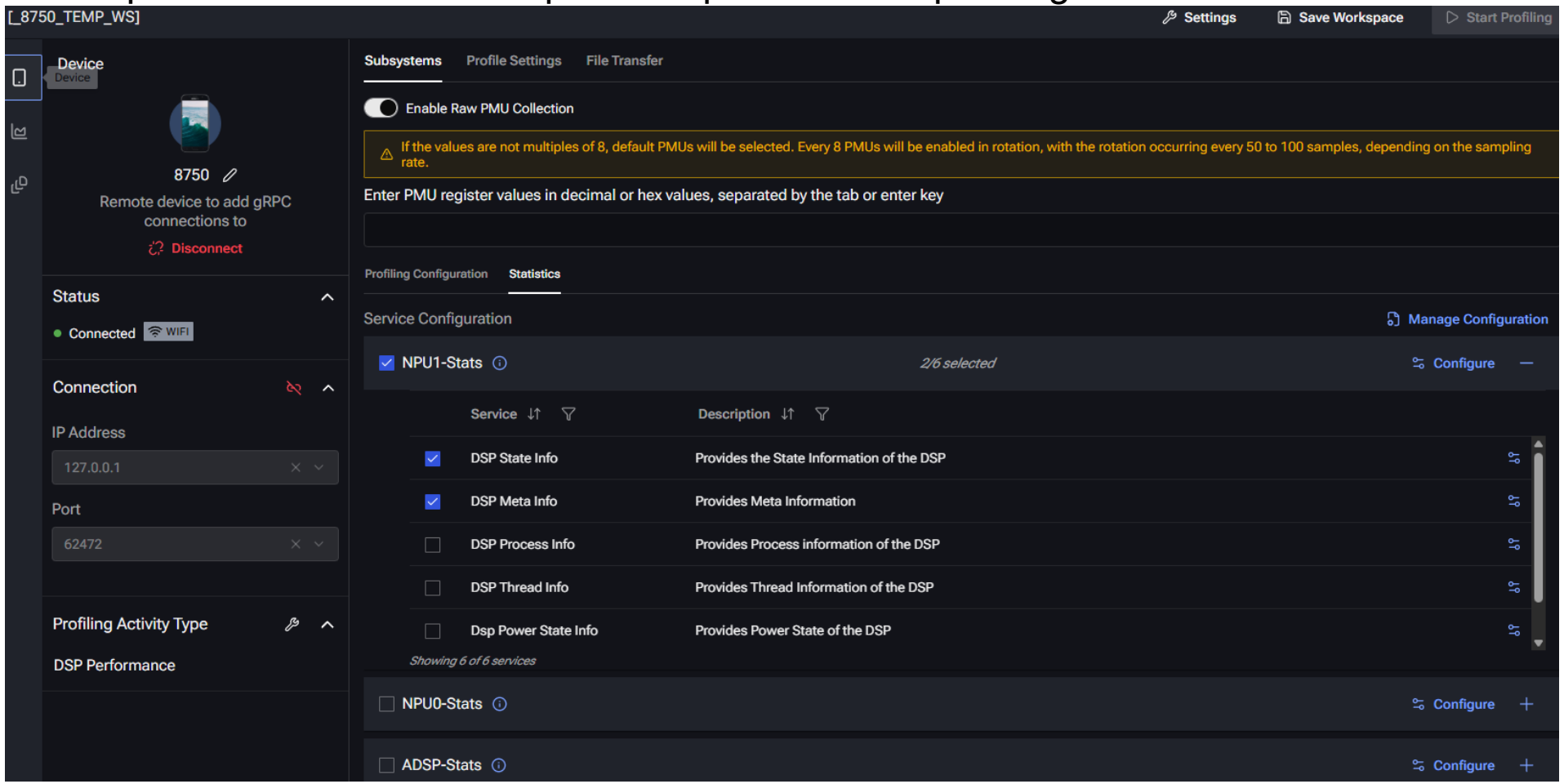
DSP performance

- Metrics will be selected automatically depending on parameter configuration. A raw data file will be generated and parsed for analysis after the profiling session.
- Set the “Profiling Activity Type” after connection to a device is successfully established:



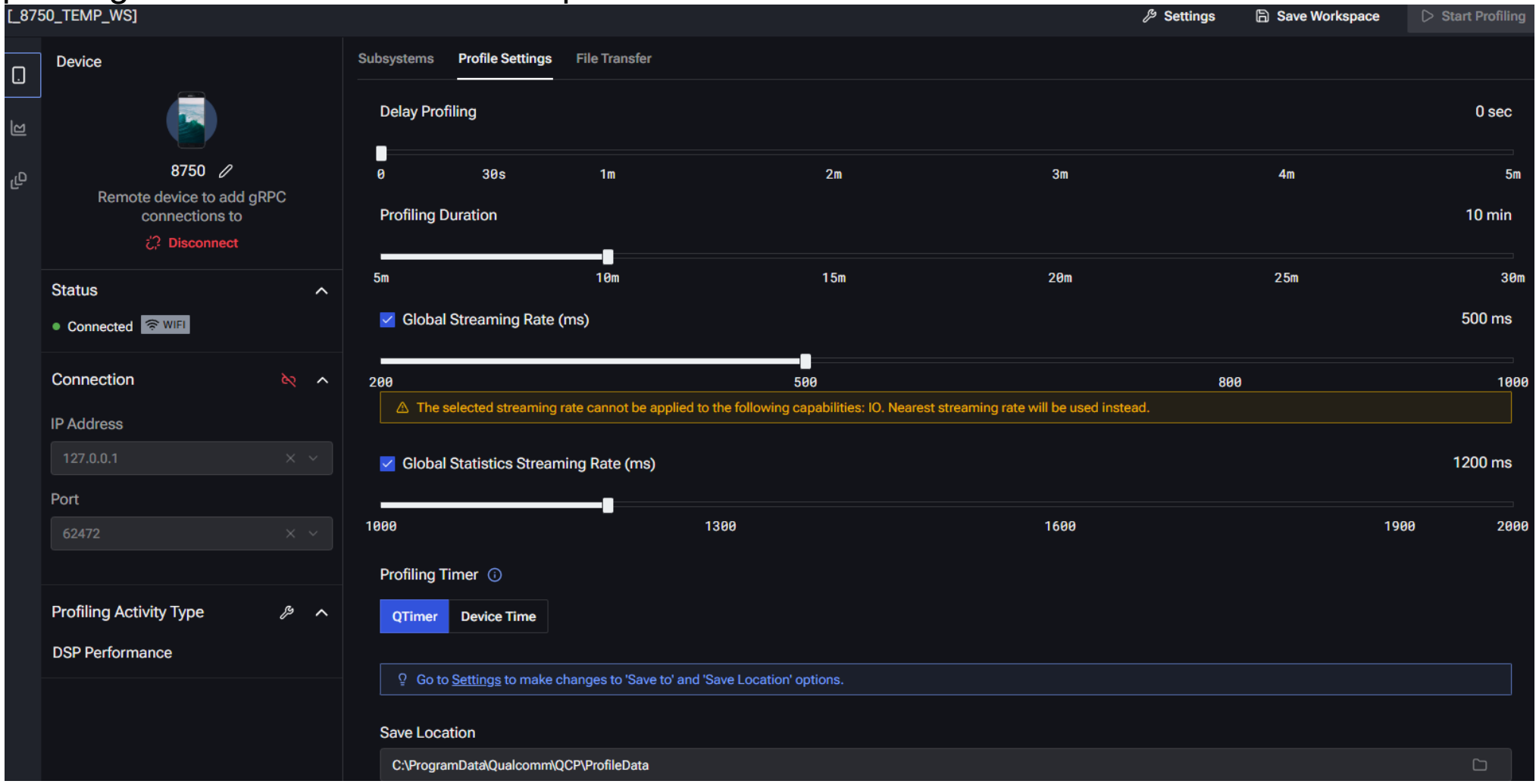
DSP performance (cont.)

- 1. The capabilities the device supports for DSP performance are shown along with the parameter configurations per capability. Select the desired capability and configure its parameters. Only one capability and its parameters are needed per DSP performance profiling session.



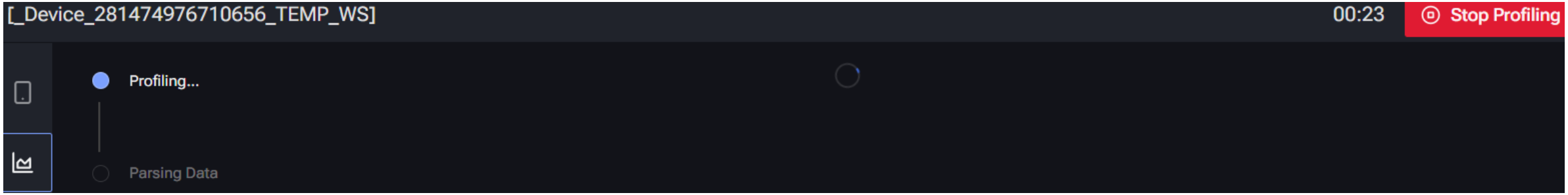
DSP performance (cont.)

- 2. Click the **Profile Settings** tab. Make selections for delay profiling, profiling duration, streaming rate, and profiling timer. Choose to save the profiled data to a file and/or a CSV.



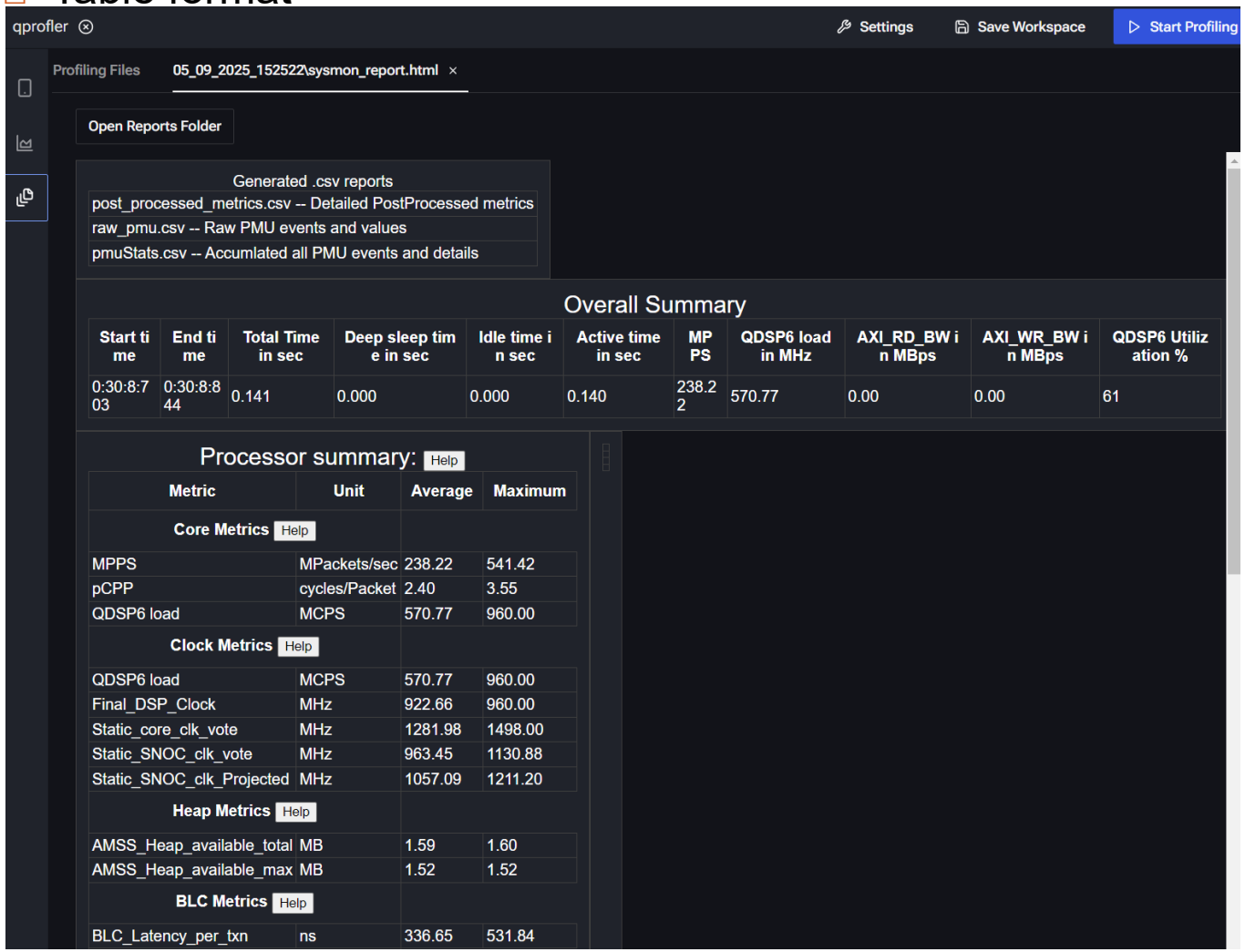
DSP performance (cont.)

- 3. Click **Start Profiling** to start the profiling session. A count down timer will appear to show the time remaining for the profiling session. The DSP Performance Profiling View displays once profiling starts.
  - Wait until the selected profiling time elapses or click **Stop Profiling** to end the profiling session at any time.



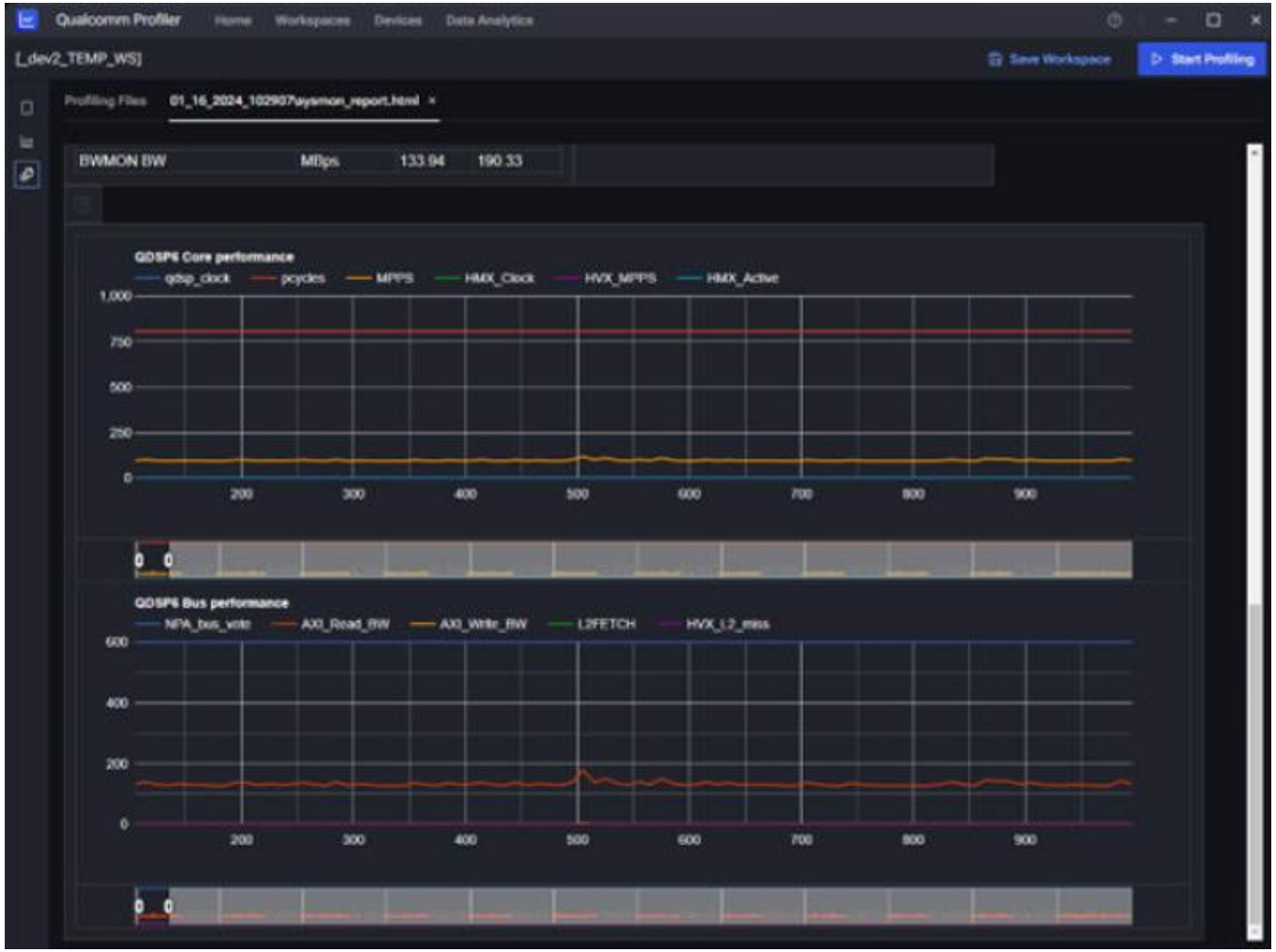
DSP performance (cont.)

- 4. Gathered DSP data can be viewed in two ways:
  - Table format



DSP performance (cont.)

- 4. Gathered DSP data can be viewed in two ways:
  - Chart format



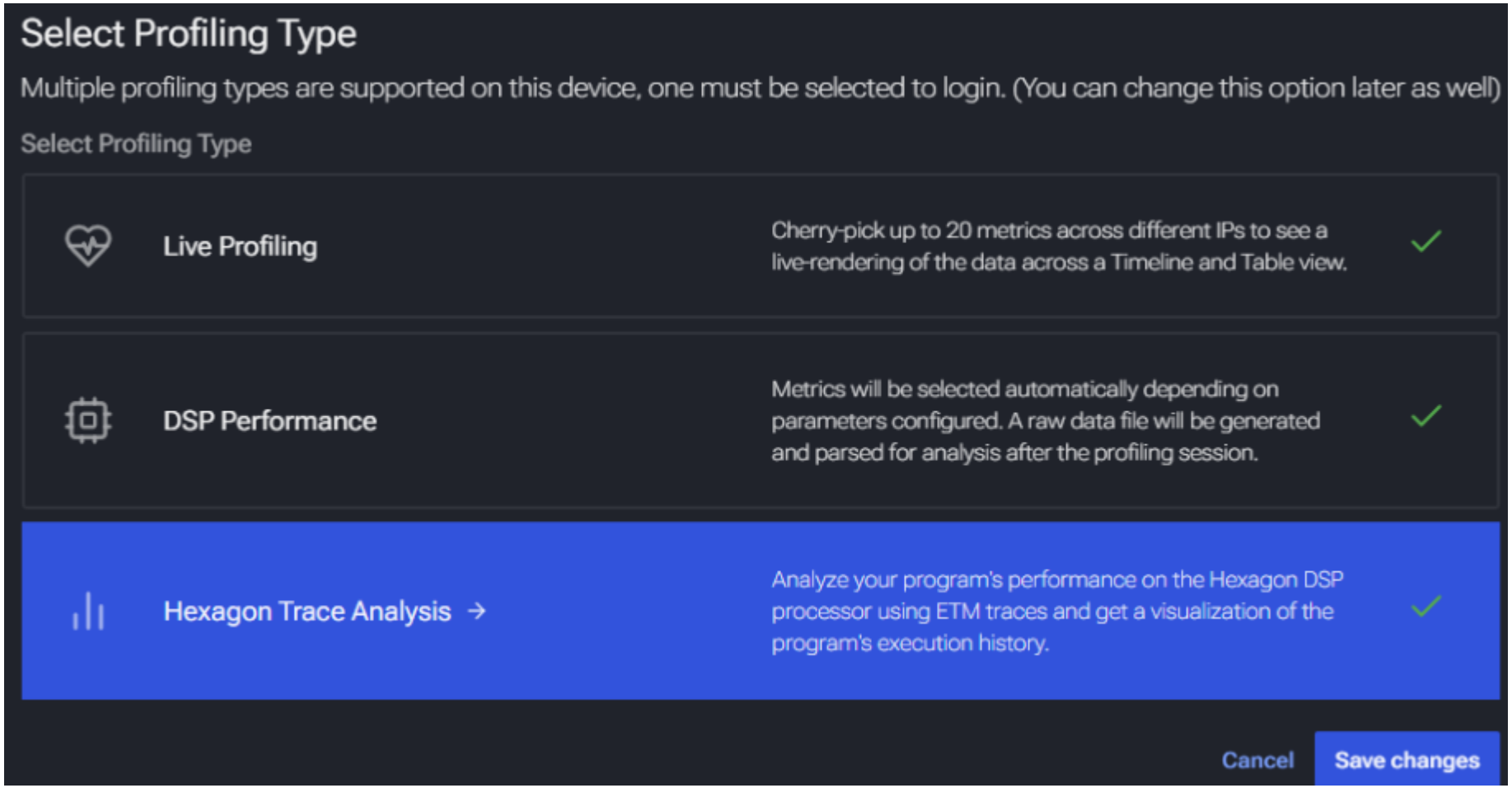
Hexagon Trace Analysis

The Qualcomm Profiler GUI can be used for Hexagon Trace Analysis（HexTA）.

- Prerequisites:
  - Hexagon SDK 5.5.0.1
  - Hexagon Tools 8.8.02 or above
  - Python 3.11.2
  - Qualcomm Profiler

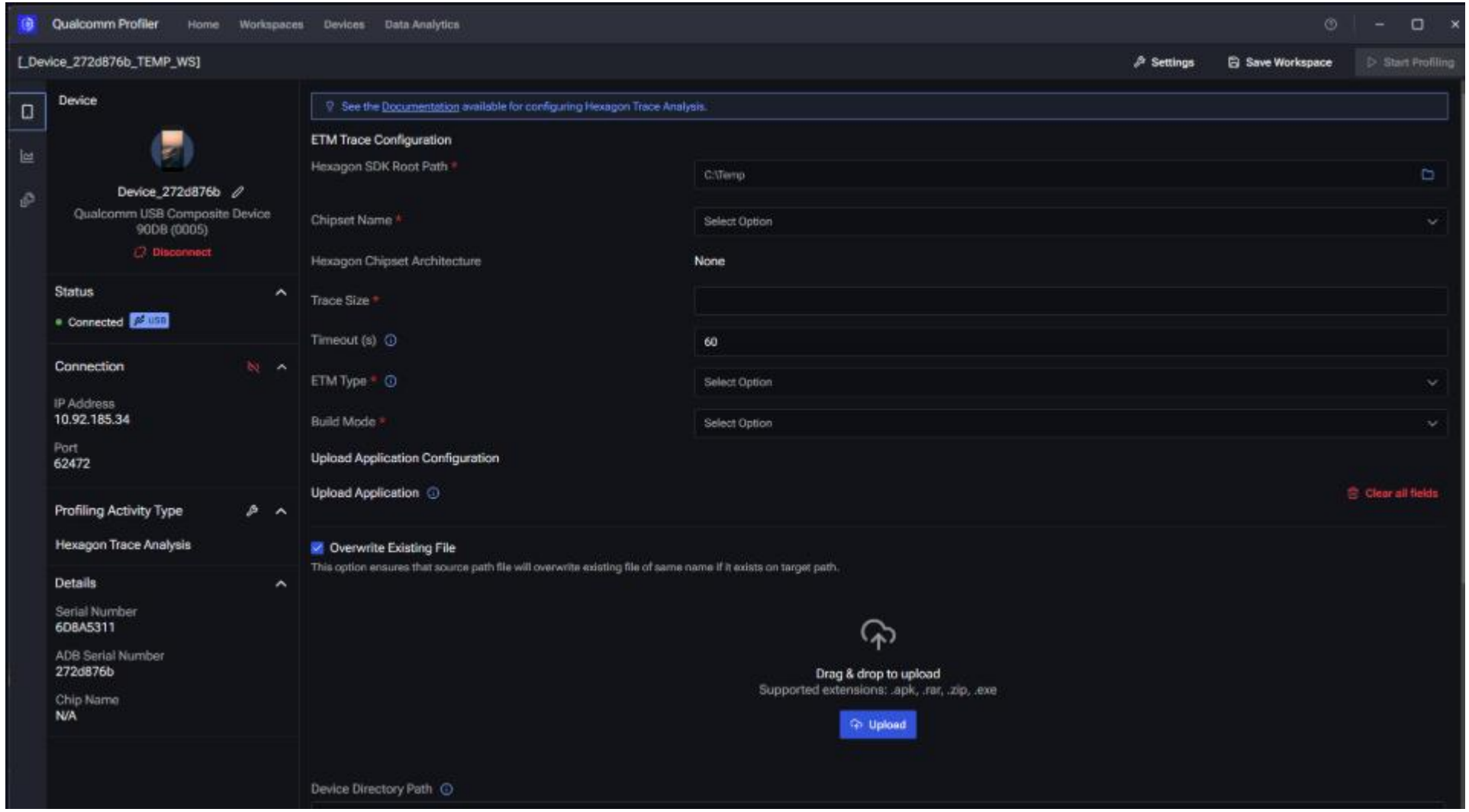
Hexagon Trace Analysis (cont.)

- Steps for Running Hexagon Trace Analysis:
  - 1.Connect to a device.
  - 2.Select Hexagon Trace Analysis as Profiling Type.



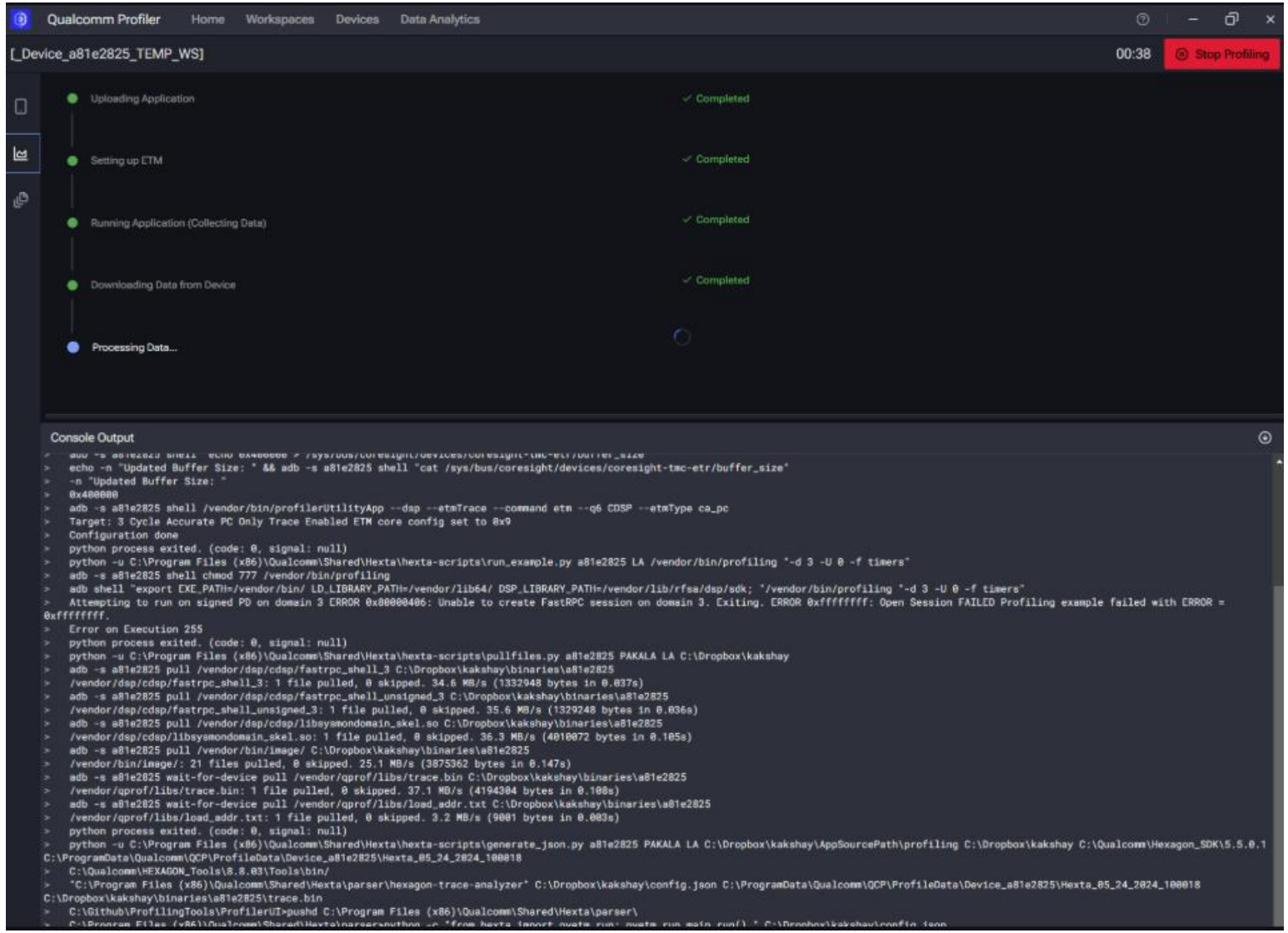
Hexagon Trace Analysis (cont.)

- 3.Enter all required fields for the Hexagon Trace Analysis form.



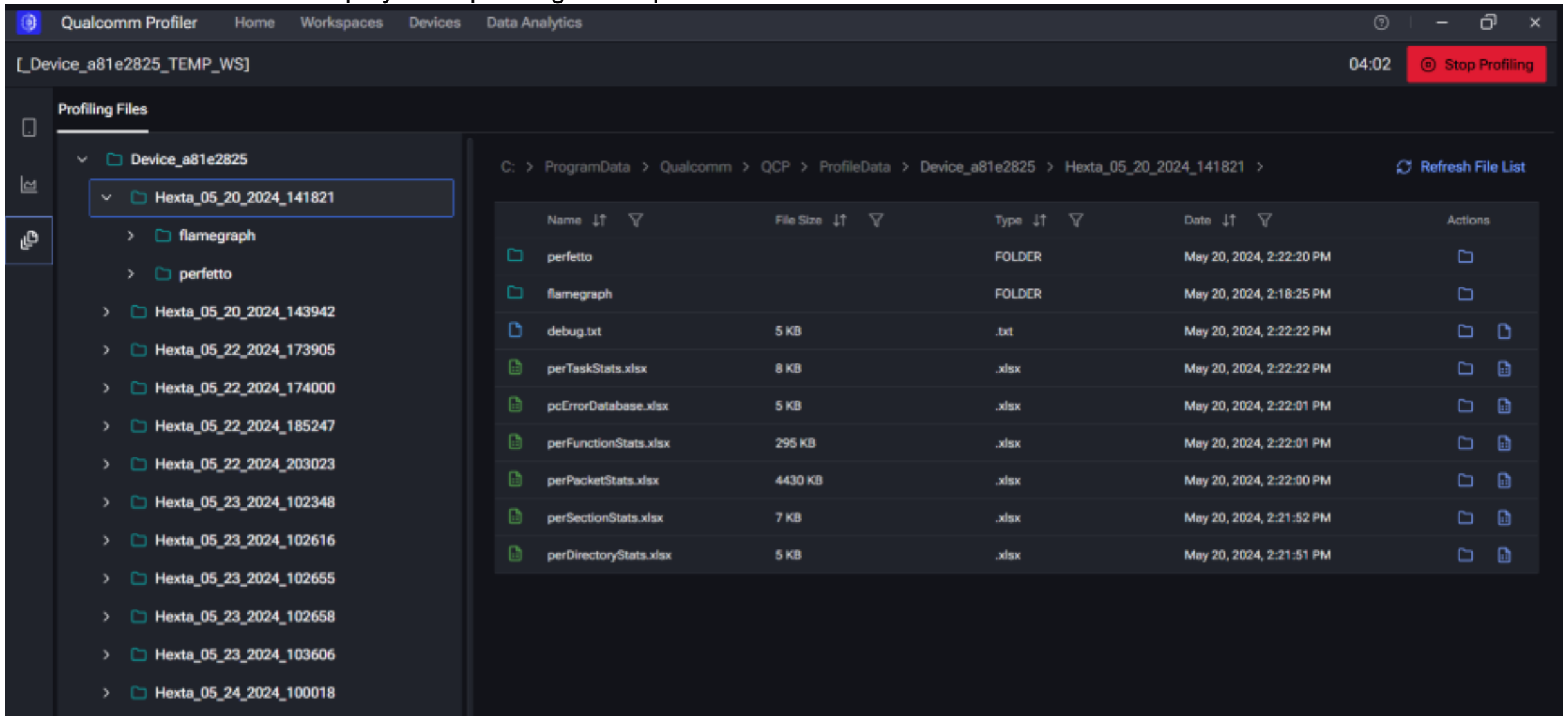
Hexagon Trace Analysis (cont.)

- 4. Click on Start Profiling.



Hexagon Trace Analysis (cont.)

- 5.The results will display once profiling is complete.



QCP Metrics and Capabilities List

Performance metrics

- Subsections of this chapter provide summaries of performance metrics.
  For more information on performance metrics, refer to the ICD HTML file:
  - For Windows (x86): **C:\Program Files (x86)\Qualcomm\Shared\Prof_Ext**
  - For Linux: /opt/qcom/Shared/Prof_Ext/
  - For WoS: C:\Program Files(Arm)\Qualcomm\Shared\Prof_Ext\

NSP performance metrics

ID	Name	Unit	Description
0x1000/4096	MPPS	MPackets/se c	Total packets executed per second
0x1001/4097	QDSP6 load	MCPS	QDSP6 load in millions of clock cycles per second
0x1002/4098	QDSP6 utilization	Percentage	QDSP6 core clock utilization
0x1003/4099	pCPP	cycles	Processor cycle per packet (pCPP)

NSP performance metrics

ID	Name	Unit	Description
0x102D/4141	AXI 128 Byte read request	MBps	128-byte line read requests issued by primary AXI master
0x102E/4142	AXI 128Byte write request	MBps	128-byte line write requests issued by primary AXI master

HLOS (Android and QNX) performance metrics

ID	Name	Unit	Description
0x1200/4608	CPU 0 Load	%	CPU Core0 % utilization
0x1201/4609	CPU 1 Load	%	CPU Core1 % utilization

DDR bandwith performance metrics

ID	Name	Unit	Description
0x1600/5632	GPU BW at DDR	MBps	GPU initiated bandwidth

Profiling capabilities

- Profiling capabilities are device dependent.

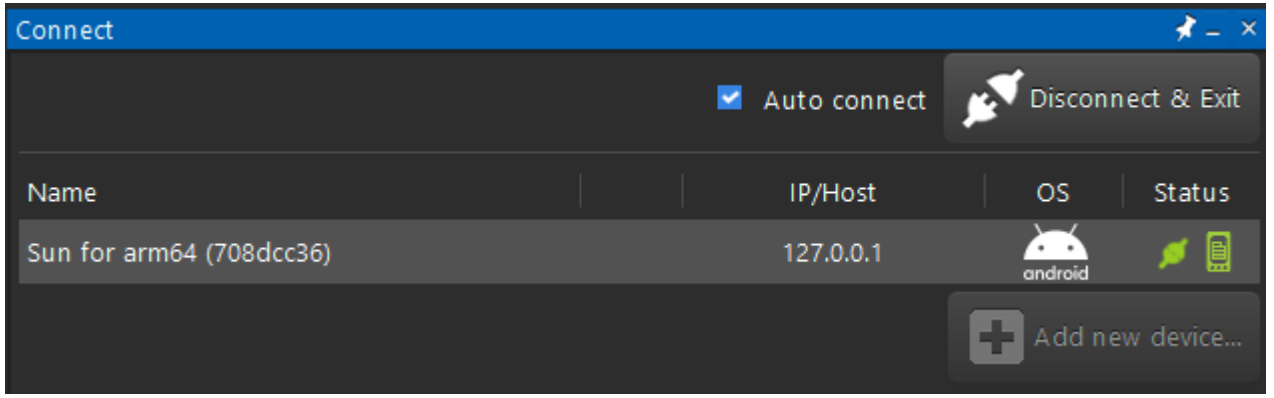
Command	Description	Android
adsp-dsp-metrics	aDSP processor metrics	Yes
ndsp-dsp-metrics	cDSP processor metrics	Yes
sdsp-dsp-metrics	sDSP processor metrics	Yes
nsp1-dsp-metrics	CSDP1 processor metrics	Yes
gdsp0-dsp-metrics	GDSP0 processor metrics	Yes
gdsp1-dsp-metrics	GDSP1 processor metrics	Yes
apps-proc-cpu-metrics	CPU process metrics	Yes
proc-gpu-specific-metrics	GPU processor metrics	Yes
apps-proc-process-metrics	Process profiling metrics	Yes
bw-profiler-ddr-metrics	DDR profiling metrics	Yes
apps-proc-ddr-metrics	DDR profiling metrics	No
adsp-dsp-stats	aDSP processor stats	Yes
cdsp-dsp-stats	cDSP processor stats	Yes
sdsp-dsp-stats	sDSP processor stats	Yes
cdsp1-dsp-stats	cDSP1 processor stats	Yes

Command	Description	Android
gdsp0-dsp-stats	gDSP0 processor stats	Yes
gdsp1-dsp-stats	gDSP1 processor stats	Yes
apps-proc-cpu-ddr-bw-metrics	CPU DDR BW metrics	No
apps-proc-io-metrics	IO metrics	Yes
apps-proc-mem-metrics	Memory metrics	Yes
wos-apps-proc-cpu-metric	CPU process metrics	No
wos-apps-proc-thermal-metrics	Thermal Metrics	No

Functionality Comparison from Snapdragon Profiler

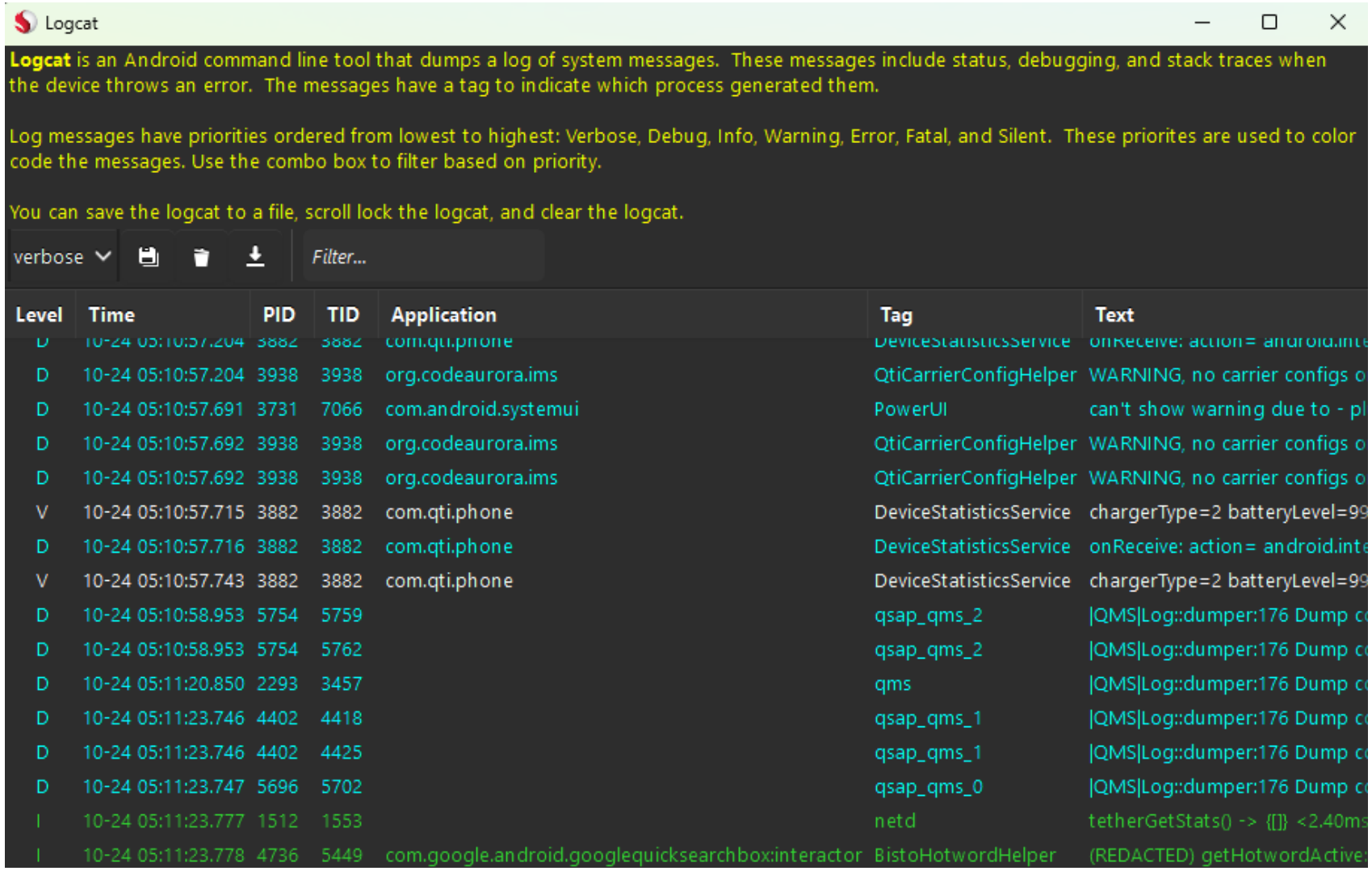
Snapdragon® Profiler is profiling software that runs on the Windows, Mac, and Linux platforms and connects with devices powered by Snapdragon processors. Snapdragon Profiler allows developers to analyze CPU, GPU, DSP, memory, power, thermal, and network data, so you can find and fix performance bottlenecks.

- For GUI usage, customer can choose different tool by purpose and try.
- C:\Program Files\Qualcomm\Snapdragon Profiler**SnapdragonProfiler.**



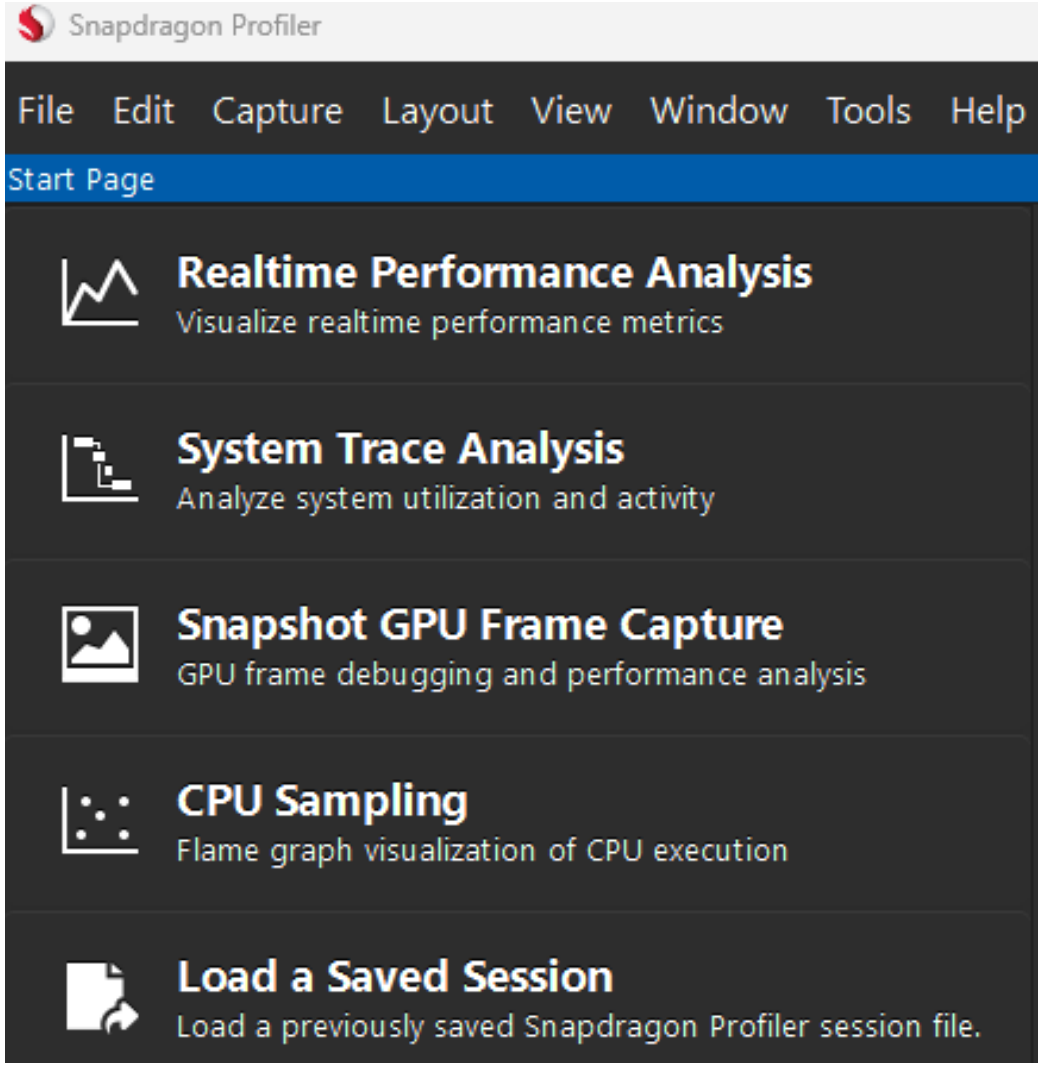
Snapdragon Profiler（SDP）Functionality

- Tools->logcat

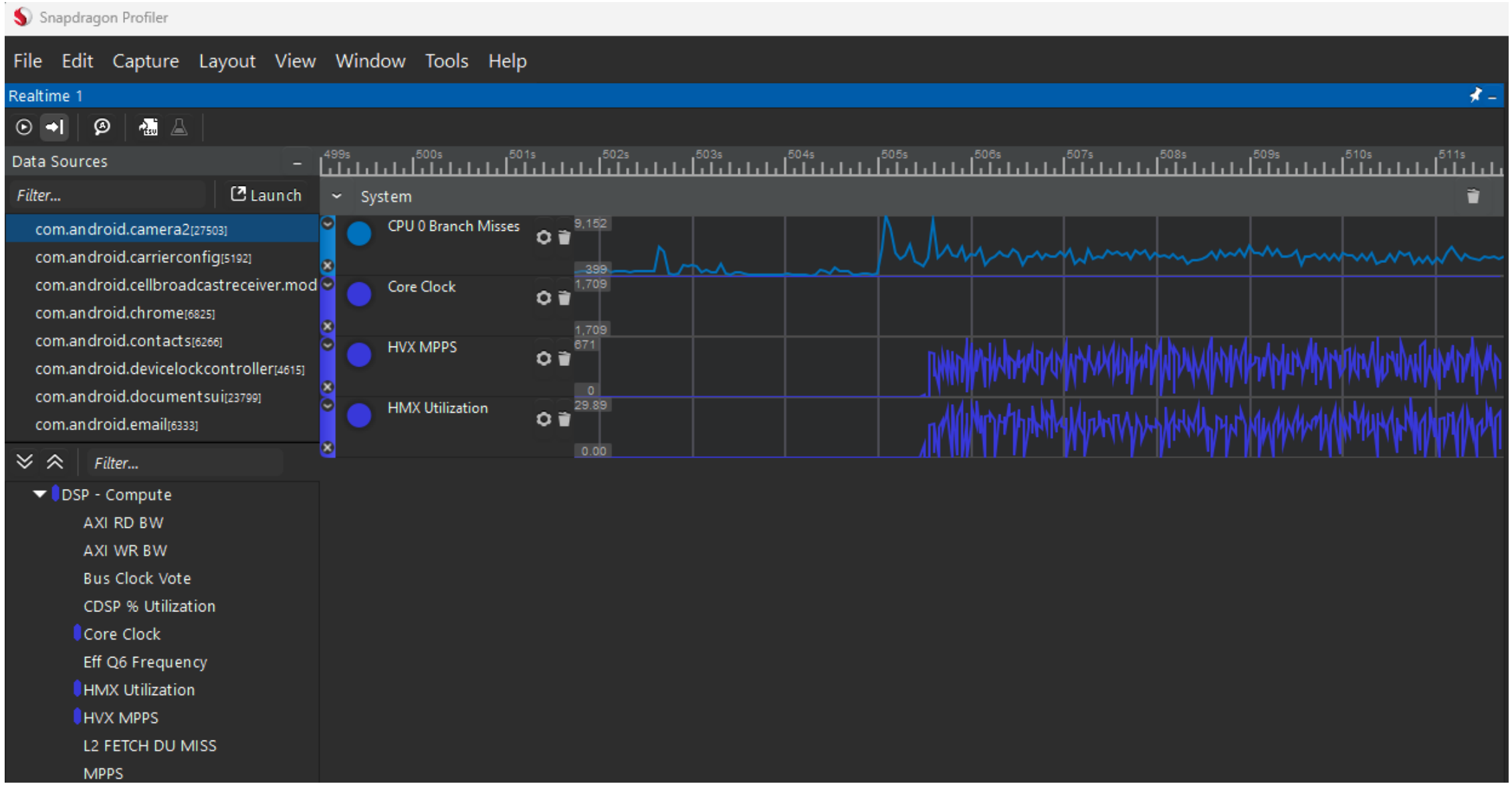


SDP Data capture modes

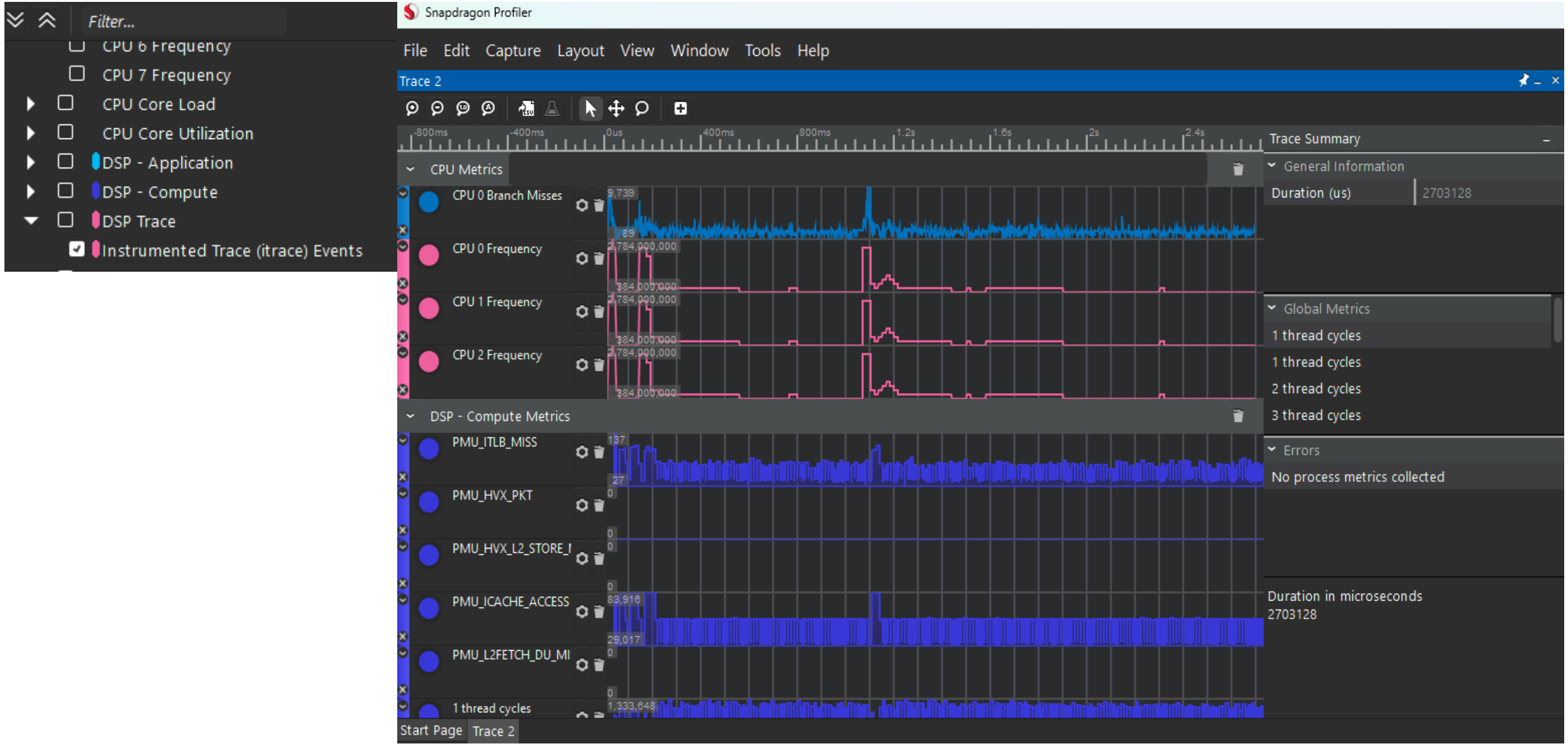
- Snapdragon Profiler provides four data capture modes offering different views of device performance: Realtime, Trace Capture, Snapshot Capture, and Sampling Capture
- **Realtime**
  - Realtime view makes it easy to correlate system resource usage at runtime.
    - Analyze CPU, GPU, DSP, memory, power, thermal, and network data metrics.
    - Select from over 150 different hardware performance counters in 22 categories
- **Trace Capture**
  - Trace Capture mode allows you to visualize kernel and system events on a timeline.
    - Analyze low-level system events across the CPU, GPU, and DSP
    - View CPU scheduling and GPU driver instrumentation with OpenGL ES, Vulkan, OpenCL, or DirectX
- Snapshot Capture
  - Snapshot Capture mode allows you to capture and debug a rendered frame from an OpenGL ES, Vulkan, or DirectX application.
- **Sampling Capture**
  - Sampling Capture mode allows you to analyze CPU performance at a function or module level through a ffamegraph visualization.



SDP Data capture modes —— Realtime

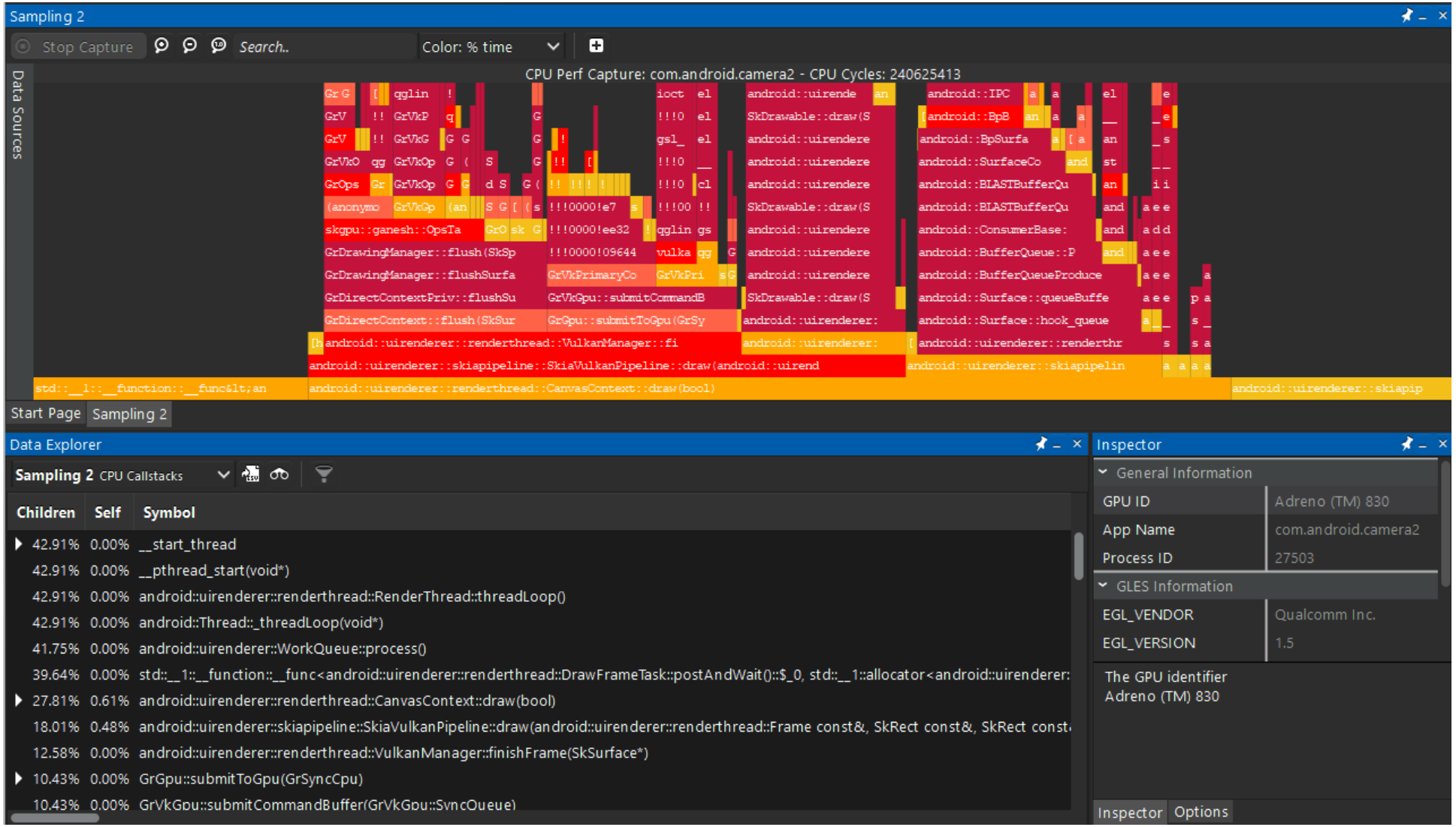


SDP Data capture modes —— Trace



SDP Data capture modes —— Sampling

- Sampling Capture mode, shown in Figure 10, captures a call graph for an application running on Snapdragon powered devices.



Section 4

References

Documents and Standards

Title	Number
Qualcomm Technologies, Inc.	
Title	
Standards	
Title	Standard Number (June 2022)
Resources	

Acronyms and Terms

Acronym or Term	Definition

Questions?

For additional information or to submit technical questions, go to: https://www.qualcomm.com/support

LEGAL INFORMATION

Your access to and use of this material, along with any documents, software, specifications, reference board files, drawings, diagnostics and other information contained herein (collectively this "Material"), is subject to your (including the corporation or other legal entity you represent, collectively "You" or "Your") acceptance of the terms and conditions ("Terms of Use") set forth below. If You do not agree to these Terms of Use, you may not use this Material and shall immediately destroy any copy thereof.

1) Legal Notice.

This Material is being made available to You solely for Your internal use with those products and service offerings of Qualcomm Technologies, Inc. ("**Qualcomm Technologies**"), its affiliates and/or licensors described in this Material, and shall not be used for any other purposes. If this Material is marked as "**Qualcomm Internal Use Only**", no license is granted to You herein, and You must immediately (a) destroy or return this Material to Qualcomm Technologies, and (b) report Your receipt of this Material to qualcomm.support@qti.qualcomm.com. This Material may not be altered, edited, or modified in any way without Qualcomm Technologies' prior written approval, nor may it be used for any machine learning or artificial intelligence development purpose which results, whether directly or indirectly, in the creation or development of an automated device, program, tool, algorithm, process, methodology, product and/or other output. Unauthorized use or disclosure of this Material or the information contained herein is strictly prohibited, and You agree to indemnify Qualcomm Technologies, its affiliates and licensors for any damages or losses suffered by Qualcomm Technologies, its affiliates and/or licensors for any such unauthorized uses or disclosures of this Material, in whole or part.

Qualcomm Technologies, its affiliates and/or licensors retain all rights and ownership in and to this Material. No license to any trademark, patent, copyright, mask work protection right or any other intellectual property right is either granted or implied by this Material or any information disclosed herein, including, but not limited to, any license to make, use, import or sell any product, service or technology offering embodying any of the information in this Material.

THIS MATERIAL IS BEING PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESSED, IMPLIED, STATUTORY OR OTHERWISE. TO THE MAXIMUM EXTENT PERMITTED BY LAW, QUALCOMM TECHNOLOGIES, ITS AFFILIATES AND/OR LICENSORS SPECIFICALLY DISCLAIM ALL WARRANTIES OF TITLE, MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, SATISFACTORY QUALITY, COMPLETENESS OR ACCURACY, AND ALL WARRANTIES ARISING OUT OF TRADE USAGE OR OUT OF A COURSE OF DEALING OR COURSE OF PERFORMANCE. MOREOVER, NEITHER QUALCOMM TECHNOLOGIES, NOR ANY OF ITS AFFILIATES AND/OR LICENSORS, SHALL BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY EXPENSES, LOSSES, USE, OR ACTIONS HOWSOEVER INCURRED OR UNDERTAKEN BY YOU IN RELIANCE ON THIS MATERIAL.

Certain product kits, tools and other items referenced in this Material may require You to accept additional terms and conditions before accessing or using those items.

Technical data specified in this Material may be subject to U.S. and other applicable export control laws. Transmission contrary to U.S. and any other applicable law is strictly prohibited.

Nothing in this Material is an offer to sell any of the components or devices referenced herein.

This Material is subject to change without further notification.

In the event of a conflict between these Terms of Use and the *Website Terms of Use* on www.qualcomm.com , the *Qualcomm Privacy Policy* referenced on www.qualcomm.com, or other legal statements or notices found on prior pages of the Material, these Terms of Use will control. In the event of a conflict between these Terms of Use and any other agreement (written or click-through, including, without limitation any non-disclosure agreement) executed by You and Qualcomm Technologies or a Qualcomm Technologies affiliate and/or licensor with respect to Your access to and use of this Material, the other agreement will control.

These Terms of Use shall be governed by and construed and enforced in accordance with the laws of the State of California, excluding the U.N. Convention on International Sale of Goods, without regard to conflict of laws principles. Any dispute, claim or controversy arising out of or relating to these Terms of Use, or the breach or validity hereof, shall be adjudicated only by a court of competent jurisdiction in the county of San Diego, State of California, and You hereby consent to the personal jurisdiction of such courts for that purpose.

2) Trademark and Product Attribution Statements.

Qualcomm is a trademark or registered trademark of Qualcomm Incorporated. Arm is a registered trademark of Arm Limited (or its subsidiaries) in the U.S. and/or elsewhere. The Bluetooth® word mark is a registered trademark owned by Bluetooth SIG, Inc. Other product and brand names referenced in this Material may be trademarks or registered trademarks of their respective owners.

Snapdragon and Qualcomm branded products referenced in this Material are products of Qualcomm Technologies, Inc. and/or its subsidiaries. Qualcomm patented technologies are licensed by Qualcomm Incorporated.