Capture a system trace on the command line

The `systrace` command invokes the Systrace tool (/topic/performance/tracing), which allows you to collect and inspect timing information across all processes running on your device at the system level.

This document explains how to generate Systrace reports from the command line. On devices running Android 9 (API level 28) or higher, you can also generate Systrace reports using the System Tracing system app (/topic/performance/tracing/on-device).

In order to run `systrace`, complete the following steps:

1. From Android Studio, download and install the latest Android SDK Tools (/studio/intro/update#sdk-manager).
2. Install Python (http://www.python.org/) and include it in your workstation's `PATH` environment variable.
3. Add `*android-sdk*/platform-tools/` to your `PATH` environment variable. This directory contains the Android Debug Bridge binary (adb), which is called by the `systrace` program.
4. Connect a device running Android 4.3 (API level 18) or higher to your development system using a USB debugging connection (/tools/device#setting-up).

The `systrace` command is provided in the Android SDK Tools package and is located in `*android-sdk*/platform-tools/systrace/`.

Syntax

To generate the HTML report for app, you need to run `systrace` from the command line using the following syntax:

```
$ python systrace.py [options] [categories]
```

For example, the following command calls `systrace` to record device activity and generate a HTML report named `mynewtrace.html`. This list of categories is a reasonable default list for most devices.

```
$ python systrace.py -o mynewtrace.html sched freq idle am wm gfx view \
    binder_driver hal dalvik camera input res
```

Tip: If you want to see the names of tasks in the trace output, you *must* include the `sched` category in your command parameters.

To view the list of categories that your connected device supports, run the following command:

```
$ python systrace.py --list-categories
```

If you don't specify any categories or options, `systrace` generates a report that includes all available categories and uses default settings. The categories available depend on the connected device you're using.

Global options

Global options	Description
`-h | --help`	Show the help message.
`-l | --list-categories`	Lists the tracing categories available to your connected device.

Commands and command options

Commands and options	Description
`-o` *`file`*	Write the HTML trace report to the specified ***file***. If you don't specify this option, `systrace` saves your report to the same directory as `systrace.py` and names it `trace.html`.
`-t` *`N`* `| --time=`*`N`*	Trace device activity for ***N*** seconds. If you don't specify this option, `systrace` prompts you to end the trace by pressing the Enter key from

	the command line.
`-b N | --buf-size=N`	Use a trace buffer size of ***N*** kilobytes. This option lets you limit the total size of the data collected during a trace.

`-k functions`
`| --ktrace=functions`
	Trace the activity of specific kernel functions, specified in a comma-separated list.

`-a app-name`
`| --app=app-name`
	Enable tracing for apps, specified as a comma-separated list of process names (/guide/topics/manifest/application-element#proc). The apps must contain tracing instrumentation calls from the `Trace` (/reference/android/os/Trace) class. You should specify this option whenever you profile your app—many libraries, such as `RecyclerView` (/reference/androidx/recyclerview/widget/RecyclerView), include tracing instrumentation calls that provide useful information when you enable app-level tracing. For more information, see Define custom events (/topic/performance/tracing/custom-events).
	To trace all apps on a device running Android 9 (API level 28) or higher, pass the wildcard character `"*"`, including the quotation marks.

`--from-file=file-path`	Create an interactive HTML report from a file, such as TXT files that include raw trace data, instead of running a live trace.

`-e device-serial`
`|`
`--serial=device-serial`
	Conduct the trace on a specific connected device, identified by its device serial number (/studio/command-line/adb#devicestatus).

`categories`	Include tracing information for the system processes you specify, such as `gfx` for system processes that render graphics. You can run `systrace` with the `-l` command to see a list of services available to your connected device.

Last updated 2020-01-13.