

## CMD-vs-ENTRYPOINT

Dockerfile:

```
FROM debian:wheezy ENTRYPOINT ["/bin/ping"] CMD ["localhost"]
```

Execution: Docker build -t test . Docker run -d test Docker run -it test google.com

Dockerfile: FROM debian:wheezy CMD ["/bin/ping", "localhost"]

Execution: Docker build -t test2 . Docker run test2 Docker run test2 bash

The ENTRYPOINT specifies a command that will always be executed when the container starts. The CMD specifies arguments that will be fed to the ENTRYPOINT.

Both CMD and ENTRYPOINT instructions define what command gets executed when running a container. There are few rules that describe their co-operation. 1. Dockerfile should specify at least one of CMD or ENTRYPOINT commands. 2. ENTRYPOINT should be defined when using the container as an executable. 3. CMD should be used as a way of defining default arguments for an ENTRYPOINT command or for executing an ad-hoc command in a container. 4. CMD will be overridden when running the container with alternative arguments

```
CMD      No CMD      error, not allowed      CMD ["exec_cmd", "p1_cmd"]
exec_cmd p1_cmd      CMD ["p1_cmd", "p2_cmd"]  p1_cmd p2_cmd      CMD
exec_cmd p1_cmd  /bin/sh -c exec_cmd p1_cmd
```

```
-- ENTRYPOINT exec_entry p1_entry      No CMD  /bin/sh -c exec_entry
p1_entry      CMD ["exec_cmd", "p1_cmd"]  /bin/sh -c exec_entry p1_entry
      CMD ["p1_cmd", "p2_cmd"]  /bin/sh -c exec_entry p1_entry      CMD
exec_cmd p1_cmd  /bin/sh -c exec_entry p1_entry
```

```
-- ENTRYPOINT ["exec_entry", "p1_entry"]      No CMD  exec_entry p1_entry
      CMD ["exec_cmd", "p1_cmd"]  exec_entry p1_entry exec_cmd p1_cmd
CMD ["p1_cmd", "p2_cmd"]  exec_entry p1_entry p1_cmd p2_cmd      CMD
exec_cmd p1_cmd  exec_entry p1_entry /bin/sh -c exec_cmd p1_cmd
```