

How to redirect stdout from child processes to supervisord

#Docker (/#tech;Docker) #Supervisord (/#tech;Supervisord)

January 8, 2015

Supervisor (<http://supervisord.org/>)'s supervisord process captures stdout of the child processes it spawns and can be configured (<http://supervisord.org/logging.html#child-process-logs>) to write it to log files or syslog. On the other hand, Supervisor is often used to manage processes running in a Docker (<https://www.docker.com/>) container. Docker has a similar feature to capture stdout from the root process in a container. These logs can then be viewed using the `docker logs` (<https://docs.docker.com/reference/commandline/cli/#logs>) command. When Supervisor is used inside a Docker container, one may therefore want to configure supervisord such that it redirects stdout of its child processes to its own stdout so that the logs of the child processes can be collected by Docker.

This can be achieved using the following options in the relevant `[program:x]` (<http://supervisord.org/configuration.html#program-x-section-settings>) section(s) of the Supervisor configuration file:

```
stdout_logfile=/dev/stdout
stdout_logfile_maxbytes=0
```

Note that since these settings are configured at the program level, it is possible to enable stdout redirection selectively for certain child processes only.

The configuration shown above works as follows:

- `/dev/stdout` is a symlink to `/proc/self/fd/1`. When a process opens that file, the system actually clones file descriptor #1 (stdout) of that process. Using this as `stdout_logfile` therefore causes supervisord to redirect the program's stdout to its own stdout (from where it will be captured by Docker).
- `stdout_logfile_maxbytes=0` disables log file rotation. Obviously, log file rotation is not meaningful for stdout. Furthermore, not specifying this option will result in the following error:

```
[Errno 29] Illegal seek
```

The reason is the default value for `stdout_logfile_maxbytes` is 50MB and supervisord is not smart enough to detect that the specified log file is not a regular file.

Of course the same technique can be applied to redirect stderr as well. This requires the following options:

```
stderr_logfile=/dev/stderr  
stderr_logfile_maxbytes=0
```