

CoreDump

This wiki show cases how to make sure your httpd generates Core Dumps on different platforms.

Generally there are two things to make sure:

- Generating coredumps is allowed
- Generating coredumps is possible

The first is usually controlled by an Operating System facility which either globally or locally limits this ability.

The latter can either be controlled by the OS, or to a certain extend by the daemon. In Apache httpd we have [CoreDumpDirectory](#) is a hint to the daemon to [chdir\(2\)](#) to a directory where it can leave behind a core file.

This is, because daemons usually run in / where they do not have write permissions.

- [Linux](#)
- [FreeBSD](#)
- [Solaris](#)

Linux

We can extend `apachectl` to enable coredumps by writing:

```
ulimit -c unlimited
```

Note that this is what controls core dump settings on a user/session base. For system wide settings see: `/etc/security/limits.conf`, respectively `limits.conf(5)`.

To help with [CoreDumpDirectory](#), see [core\(5\)](#) how generate a `core_pattern`, which you can set via:

```
sysctl -w kernel.core_pattern=/some/core/pattern
```

FreeBSD

Depending on the shell, we can either use `ulimit -c unlimited` if we're using [sh\(1\)](#), or, if we use [csh\(1\)](#):

```
limit coredumpsize unlimited
```

Note that this is what controls core dump settings on a user/session base. For system wide settings see: `/etc/login.conf`, respectively [login.conf\(5\)](#), or set the `kernel.coredumps` to 1:

```
sysctl kern.coredumps=1
```

To help with [CoreDumpDirectory](#), see [core\(5\)](#) how generate a `corefile` which you can set via:

```
sysctl kern.corefile=/some/core/pattern
```

Solaris

Everything of the above can be controlled via [coreadm\(1M\)](#), see this [How To](#) for a comprehensive list of examples.