

# DPDK

**Tungsten Fabric** datapath vRouter supports high performance DPDK mode of operations, for which Tungsten Fabric vRouter utilises OpenSource DPDK libraries.

Standard Tungsten Fabric build process utilises [contrail-dpdk](#) which itself is a fork of [dpdk source](#). however it includes extra bug fixes that are not yet upstream'd, enabling tungsten fabric to provide certain functionality which might not be available with upstream dpdk releases. Recently community is looking forward to support and enable build process to be able to refer to a pre-compiled dpdk binary, which can be built out of upstream release code of [dpdk](#) or any other third party compiled dpdk library.

## Alternate Build Process

This section tries to outline the changes done as part of [TFB-1445](#), to enable Tungsten Fabric build with a pre-compiled dpdk library. This pre-compiled dpdk library however needs to satisfy following conditions

- DPDK version v18.05.1
- DPDK library should have been compiled with CONFIG\_RTE\_PKTMBUF\_HEADROOM set to minimum of 256, needed by Tungsten Fabric vRouter to capture mirror data
- DPDK library may be compiled with CONFIG\_RTE\_MACHINE set to sandybridge architecture allowing build portability
- user may want to enable CONFIG\_RTE\_PORT\_STATS\_COLLECT

Once this DPDK compiled library is available, dpdk vrouter can be built using it by scons with option "**--dpdk-dir=**" pointing to the pre-compiled dpdk library.

As part of the standard Tungsten Fabric build process as well if environment variable "**DPDK\_BUILD\_DIR**" is set before triggering "**make rpm**".

Above build process expects following patches to be available in the codebase

- <https://github.com/Juniper/contrail-vrouter/commit/ce5c0f15058a15fdf040a8a28a431b5a2e62317c>
- <https://github.com/Juniper/contrail-packages/commit/44a967f6ee1ad44f2950631d96828a858e1525ef>

## Current Limitations

- missing support for VM based dev-environments, limits active development/testing on virtual environments
- missing support for CNI interfaces