Debugging NXDB

This appendix includes the following sections:

Collecting Logs and Debugging Information

To collect logs and debugging information for the external controller, enter the **show tech-support controller** command. This command also creates another tar.gz file in bootflash bootflash:Tech-NXOS_DME_logs_20160809_134358.tar.gz, which also needs to be copied as part of the tech-support collection. This file is displayed as part of the a severity 2 syslog message at the time of command execution.

```
switch(config)# terminal monitor
switch(config)# show tech-support controller > sh_tech_ctrl
2016 Aug 9 13:43:58 switch %USER-2-SYSTEM_MSG: Open NxOS Logs stored as
bootflash:Tech-NXOS_DME_logs_20160809_134358.tar.gz, please collect this
file as part of show tech support -vsh
```

To collect logs and debugging information for the switches, enter the **show tech-support vxlan** and **show tech-support vxlan platform** commands.

To collect logs and debugging information for the object store, enter the **show tech-support object-store user admin** command. The object store log is stored in a file, and the command output displays the name and location of the log file (for example, /logflash/debug_logs/objstore_20160408_101450.log).

To collect logs and debugging information for the OVSDB plugin, enter the **guestshell run sudo ovsdb-plugin tech-support** command. As the OVSDB plugin logs are collected, you can store them in a different server to retrieve them. To do so, use the **run guestshell** command to connect to the Guest Shell, copy logs/tech-support to the bootflash, and SCP from the bootflash to the location where this log can be retrieved. The log is located at /usr/local/ovsdb/ovsdbplugin_20160809-142608.tar.gz and is displayed as part of the tech-support command execution.

```
switch# guestshell run sudo ovsdb-plugin tech-support
nohup: ignoring input and appending output to 'nohup.out'
nohup: ignoring input and redirecting stderr to stdout
Data archive written to /usr/local/ovsdb/ovsdbplugin_20160809-142742.tar.gz
```

You can also collect logs and debugging information for the OVSDB plugin using SCP.

```
switch# guestshell run sudo ovsdb-plugin tech-support scp admin@172.31.145.141:/
--vrf management
```

```
nohup: ignoring input and appending output to 'nohup.out'
nohup: ignoring input and redirecting stderr to stdout
Data archive written to /usr/local/ovsdb/ovsdbplugin_20160809-142504.tar.gz
Sending archive to admin@172.31.145.141:/...
The authenticity of of host '172.31.145.141 (172.31.145.141)' can't be established.
RSA key fingerprint is bc:43:88:61:b5:27:ee:90:3b:fb:4a:b5:8c:38:ae:f9.
Are you sure you want to continue connecting(yes/no)? yes
Warning: Permanently added '172.31.145.141' (RSA) to the list of known hosts.
User Access Verification Password:
ovsdbplugin_20160809-142504.tar.gz 100%  20MB 20.0MB/s 00:01
```

To collect the OVSDB client dump from the OVSDB plugin, enter the **ovsdb-client -v dump tcp:1.1.24.1 hardware_vtep** command. The switch interface in this command should be reachable from the OVS client and should be in the same VRF on which the OVSDB plugin is running.