User Manual for 24.08.00 release

Load the diag sample applications to the target

1. Set up the TelAF application development environment referring to the TelAF user guide at Snapdragon Telematics Application Framework (TelAF) User Guide (qualcomm.com).
2. Compile the **tafDiagApp** and **tafDiagEventApp** under telaf/apps/sample with **sa525m** as the target.
3. Push the application packages to the target via **adb** tool.

We introduced the diagnostic configuration module in this release to control the specific actions of UDS, such as DEM & DCM.

Configuration module is required and generated by tafDiagGen tool.

Some additional steps for configuration module need to be done.

Refer to the following steps:

1. Get the tafDiagGen tool patch from QC side and put the patch to the 'telaf' directory
2. Patch the patch by:
    1. patch -p 1 < tafDiagGen_Tool.patch
    2. <tafDiagGen_Tool> needs to be token place with a valid name you get from QC side.
3. Setup the basic python environment
    1. python --version (required 3.8 or higher version)
    2. cd apps/tools/tafDiagGen
    3. make setup
4. Install the generated source code and the configuration file to the telaf project.
    1. make install
5. Config functional groups in components/tafDoIPStack/tafDoIP.json for functional addressing if needed.
6. Re-build TelAF or tafDiagSvc alone.
7. Push the new tafDiagSvc to target via adb tool again or reflash the telaf a/b volumes by fastboot.

Target side:

1. Set DID values and attributes with the following config tree commands:

```
config set diag/DID/f011/value/data1 81 int

config set diag/DID/f011/value/data2 82 int

config set diag/DID/f011/value/data3 83 int

config set diag/DID/f011/value/data4 84 int

config set diag/DID/f011/value/data5 85 int

config set diag/DID/f011/value/data6 86 int

config set diag/DID/f011/value/data7 87 int

config set diag/DID/f011/value/data8 88 int

config set diag/DID/f011/value/data9 89 int

config set diag/DID/f011/value/data10 79 int

config set diag/DID/a5a5/value/data1 1 int

config set diag/DID/a5a5/value/data2 2 int

config set diag/DID/f0d0/value/data1 17 int

config set diag/DID/f0d0/value/data2 34 int

config set diag/DID/f0d0/value/data3 51 int

config set diag/DID/f0d2/value/data1 17 int

config set diag/DID/f0d2/value/data2 34 int

config set diag/DID/f0d2/value/data3 51 int

config set diag/DID/f0d2/value/data4 68 int

config set diag/DID/f0d2/value/data5 85 int

config set diag/DID/f0d2/value/data6 102 int

config set diag/DID/f401/value/data1 17 int

config set diag/routine/targetFile "/data/images/update_ubi_ab.zip" string
```

2.Run sample application to handle diag service requests.

app start tafDiagApp

3.Run event sample application to create DTC data.

app start tafDiagEventApp

4. Create the directory that firmware update package will be downloaded to.

mkdir /data/images

Start UDS client on PC host

1. Setup the PC host environment to be able to run the script tool.

python3 -m pip install doipclient==1.1.1

python3 -m pip install udsoncan==1.17.1

2. Copy the script tool from **`telaf/apps/tools/diag/update_client.py`** to the home directory of the UDS client PC host.

3. Config the IP address of UDS server by changing the line 72 in the file of update_client.py based on the target's IP configuration.
DoIPClient("192.168.225.1", 513) // IP address : 192.168.225.1 logical address: 513

4. Config the location of the file to transfer by changing the line 54 in the file of update_client.py based on the absolute source path of the file.
update_file = "/home/ubuntu/data/update_ubi_ab.zip"

5. Make sure it's bidirectional ethernet connection reachable between UDS client PC host and the target.

6. Run python script from client PC.

python3 update_client.py

Then, the UDS client is running properly.

Note: Before every new run of the diag sample app on the target, the previously downloaded files under /data/images need to be removed.

Multi VLAN:
Required Apps: tafDiagSvc1, tafDiagSvc2, tafDiagApp1, tafDiagApp2.
Required python scripts: update_client1.py, update_client2.py.

Set up the TelAF application development environment same as above but from Target side follow below steps.

1. First create VLAN interfaces by pushing **tafNetIntTest.sa525.update** file in sa525m device.

   Create interfaces:

   app runProc tafNetIntTest --exe=tafNetIntTest -- createvlan 1 2 0
   app runProc tafNetIntTest --exe=tafNetIntTest -- createvlan 2 2 0

   ip link set eth0.1 down
   ip link delete eth0.1
   ip link add link eth0 name eth0.1 address 00:55:7b:b5:7d:fc type vlan id 1
   ip addr add 10.10.1.1/24 dev eth0.1
   ip link set eth0.1 up
   route add 225.0.0.1 dev eth0.1

   ip link set eth0.2 down
   ip link delete eth0.2
   ip link add link eth0 name eth0.2 address 00:55:7b:b5:7d:fe type vlan id 2
   ip addr add 10.10.2.1/24 dev eth0.2
   ip link set eth0.2 up
   route add 226.0.0.1 dev eth0.2

   In First client device:(can be done by using nmtui tool in ubuntu)

   ip link set eth0.1 down
   ip link delete eth0.1
   ip link add link eth0 name eth0.1 address 00:55:7b:b5:7d:fd type vlan id 1
   ip addr add 10.10.1.2/24 dev eth0.1
   ip link set eth0.1 up
   route add 225.0.0.1 dev eth0.1

   In Second client device:(can be done by using nmtui)
   ip link set eth0.2 down
   ip link delete eth0.2
   ip link add link eth0 name eth0.2 address 00:55:7b:b5:7d:ff type vlan id 2
   ip addr add 10.10.2.2/24 dev eth0.2
   ip link set eth0.2 up
   route add 226.0.0.1 dev eth0.2

2. Set DID values and attributes with the following config tree commands (same as above mentioned).

3. Run two diag services to serve diag requests coming from two VLAN's.

   app start tafDiagSvc1

   app start tafDiagSvc2

4. Run sample application 1 to handle diag service requests from two python UDS clients.

   app start tafDiagApp1

5. Create the directories that firmware update packages will be downloaded to for two UDS clients:
   mkdir /data/image1

   mkdir /data/image2

Make sure UDS client setup was there on two PC connected with two VLAN via switch.

1. Copy the script tool from **`telaf/apps/tools/diag/update_client1.py and telaf/apps/tools/diag/update_client2.py`** to the home

directory of the respective UDS client PC host.

2. Config the IP address of UDS server by changing the line 74 in the file of update_client1.py and update_client2.py based on the target's IP configuration. (Make sure that VLAN interfaces created IP in sa525m device, and this IP are same)
   DoIPClient("10.10.1.1", 513) // IP address : 10.10.1.1 logical address: 513
   DoIPClient("10.10.2.1",514) // IP address : 10.10.2.1 logical address: 514
3. Config the location of the file to transfer by changing the line 32 in the file of update_client1.py and update_client2.py based on the absolute source path of the file.
4. Make sure it's bidirectional ethernet connection reachable between UDS client PC host's and the target.
5. Run two python scripts from two client PC.
   python3 update_client1.py
   python3 update_client2.py

Then, the UDS client is running properly.

Note: Before every new run of the diag sample app on the target, the previously downloaded files under /data/image1 and /data/image2 need to be removed.

DID Storage Service:

1. Required Apps: tafDidStoreSvc, tafDidStoreUnitTestApp, tafDiagApp.
2. Required plugin module: tafPiDidStore.
3. Compile and push the unit test app and diag sample app packages to the target via adb tool.
4. Compile and push the plugin module to the target via adb tool.

Target side:

a. Stop the tafDidStoreSvc and install plugin module.
   app stop tafDidStoreSvc

   tafmodule install /data/tafPiDidStore.so

b. Start the tafDidStoreSvc.
   app start tafDidStoreSvc

1.Testing using unit test App.

   a. Install and run the unit test app.
      app start tafDidStoreUnitTest

   This test app will read/write and notify from the plugin module for the specific DID(0xA5A5, 0xA5A6).

2. Testing using UDS client Setup.

   Follow the above instruction for setting up UDS client.

   The UDS client will get the DID read/write from the storage via plugin.

   a. Install and run the diag sample app on target.
      app start tafDiagApp

   b. Run the command in client setup.
      python update_client.py