

O-DU High and Low Integration

This page describes O-DU High/Low pairwise testing procedure in Timer mode.

Demo Video

Your browser does not support the HTML5 video element

Delivery Status

 **DO NOT EDIT**

Only the Integration PTL should edit the delivery status table to track the release validation progress

		OTF	INF	OAM	NONRTRIC	RICP	RICAPP	O-DU	O-CU	Test Result	Notes
Deploymen t Artifacts	Docker Container										Dockerfiles are released, however, containers are not generated and te
	Helm Charts										
	Deployment scripts										
	OS Image										
E2E Flows	WLS interface ready										
	UL IQ sample request										
	UL IQ sample response										
	Config request										
	Config response										
	Start request										
	Slot indication										

1. Cloning O-DU High code

```
mkdir odu_high
cd odu_high
git clone "https://gerrit.o-ran-sc.org/r/o-du/12"
```

2. Cloning O-DU Low code

```
mkdir odu_low
cd odu_low
- git clone "https://gerrit.o-ran-sc.org/r/o-du/phy"
- Download code from "https://github.com/intel/FlexRAN"
```

3. Install O-DU High pre-requisite libraries by following <https://docs.o-ran-sc.org/projects/o-ran-sc-o-du-l2/en/latest/installation-guide.html> .
4. Install O-DU Low by following <https://docs.o-ran-sc.org/projects/o-ran-sc-o-du-phy/en/latest/index.html> .
5. Compilation Preparation

```
mkdir odu_high/l2/src/wls_lib
mkdir odu_high/l2/src/dpdk_lib

cp odu_low/phy/wls_lib/wls_lib.h odu_high/l2/src/wls_lib/
cp odu_low/dpdk-19.11/x86_64-native-linuxapp-gcc/include/rte_* odu_high/l2/src/dpdk_lib/
```



DPDK library

O-DU low instructs to build DPDK using x86_64-native-linuxapp-icc

O-DU high uses x86_64-native-linuxapp-gcc, or library will be missing.

6. Compilation

```
cd odu_high/l2/build/odu
make odu PHY=INTEL_L1 PHY_MODE=TIMER MACHINE=BIT64 MODE=FDD
```



Hardcoded Library Location

The makefile of odu assumes the required libraries is located at /opt/intel
Modification of the makefile is needed if the libraries are located at different location.

7. Execution - Bringing up L1 of O-DU Low

```
cd odu_low/phy
source ./setupenv.sh
cd odu_low/FlexRAN/l1/bin/nr5g/gnb/l1
./l1.sh -e
```

Non BBU threads in application

```
=====
==
nr5g_gnb_phy2mac_api_proc_stats_thread: [PID: 8659] binding on [CPU 0] [PRIO: 0] [POLICY: 1]
wls_rx_handler (non-rt): [PID: 8663] binding on [CPU 0]
=====
==
```

PHY>welcome to application console



Dynamic Library Path

May need to include the paths to the following libraries in the environment variable LD_LIBRARY_PATH

./odu_low/FlexRAN/libs/cpa/bin

./odu_low/phy/wls_lib

./odu_low/phy/fhi_lib/lib/build

8. Execution - Bringing up FAPI Translator of O-DU Low

```
cd odu_low/phy
source ./setupenv.sh
cd odu_low/phy/fapi_5g/bin/
./oran_5g_fapi -cfg=oran_5g_fapi.cfg
```

9. Assign virtual IP addresses as follows:

```
ifconfig <interface name>:ODU "192.168.130.81"

ifconfig <interface name>:CU_STUB "192.168.130.82"

ifconfig <interface name>:RIC_STUB "192.168.130.80"
```

10. Execution - Running CU Stub

```
cd odu_high/bin/cu_stub
./cu_stub
```

11. Execution - Running RIC Stub

```
cd odu_high/bin/ric_stub
./ric_stub
```

12. Execution - Running O-DU High

```
cd odu_high/bin/odu
export LD_LIBRARY_PATH=odu_low/phy/wls_lib/lib:$LD_LIBRARY_PATH
./odu
```