

# Introduction of **SRS UDP Socket Logging**

## Proprietary and Confidential

The information contained in this document is the proprietary and exclusive property of REIGN except as otherwise indicated. No part of this document, at any time or in any manner, directly or indirectly, may be disclosed, reproduced, stored, transmitted or used for any purpose other than the purposes set forth in this document without the prior written permission of REIGN.

This document and the information contained herein shall be maintained in confidential and secret manner, and should only be disclosed to or obtained by those who are on a “need-to-know” basis and have signed the non-disclosure agreement (“NDA”) with REIGN (the “Receiving Party”).

REIGN shall be entitled to recover any damages from any person who violated this “Proprietary and Confidential Term” in accordance with the applicable law.

## Disclaimers

All information is provided “as is” and for information purpose only, without warranty of any kind and REIGN disclaims all implied warranties, including non-infringement, merchantability, and fitness for a particular purpose. The entire risk arising out of the use of the information contained herein remains with the Receiving Party. The Receiving Party acknowledges that it has not relied on and will not rely on any of the sections nor statements regarding the accuracy of the information contained in this document and the Receiving Party will have no basis for bringing any claim for fraud. The information contained in this document is subject to change without prior notice.

## Revision History

Revision	Date	Description
V1.0	2024/12/24	First release.
V2.0	2025/01/03	Change the SRS logging method from shared memory to UDP socket.

# Contents

1	Overview	4
2	Prerequisite	4
2.1	xAPP Inside the BBU Server	4
2.2	xAPP Outside the BBU Server	4
3	Start/Stop RAN Service	5
4	Run SRS UDP Socket Tool	6
4.1	xAPP Inside the BBU Server	6
4.2	xAPP Outside the BBU Server	6
5	Definition of srs_udp_log.h	7
5.1	Structure Definitions	7

# 1 Overview

This document provides instructions for the basic operation of RAN services on REIGN CORE. It also provides steps on how to execute the SRS (Sounding Reference Signal) UDP socket tool.

## 2 Prerequisite

### 2.1 xAPP Inside the BBU Server

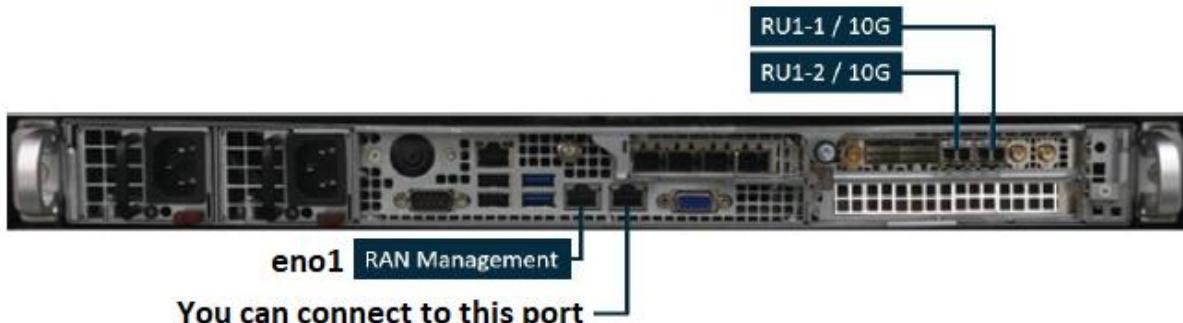
- Create a new SSH session
- Ensure that the srsUdpLogSocketIP is set to **127.0.0.1**.

```
$ vim /home/htc_l1app/HTC_L1APP/bin/nr5g/gnb/l1/netcfg_xran.xml
```

```
<!-- SRS UDP log enable -->
<srsUdpLogEnable>1</srsUdpLogEnable>
<!-- SRS UDP log destination IP -->
<srsUdpLogSocketIP>127.0.0.1</srsUdpLogSocketIP>
<!-- SRS UDP log destination Port -->
<srsUdpLogSocketPort>2498</srsUdpLogSocketPort>
```

### 2.2 xAPP Outside the BBU Server

eno1 is the interface used for RAN management, and it is not recommended to use this port to connect with your server. We recommend connecting your server to the BBU's eno2 interface.



- Create a new SSH session
- Ensure that the srsUdpLogSocketIP is set to **your server's IP** on BBU server.

EX: The IP of the port on your server that interfaces with our BBU is 192.168.190.103.

```
$ vim /home/htc_l1app/HTC_L1APP/bin/nr5g/gnb/l1/netcfg_xran.xml
```

```
<!-- SRS UDP log enable -->
<srsUdpLogEnable>1</srsUdpLogEnable>
<!-- SRS UDP log destination IP -->
<srsUdpLogSocketIP>192.168.190.103</srsUdpLogSocketIP>
<!-- SRS UDP log destination Port -->
<srsUdpLogSocketPort>2498</srsUdpLogSocketPort>
```

- Place the srs\_udp\_log\_reader folder we provided in any directory on your server and modify the folder's permissions accordingly.

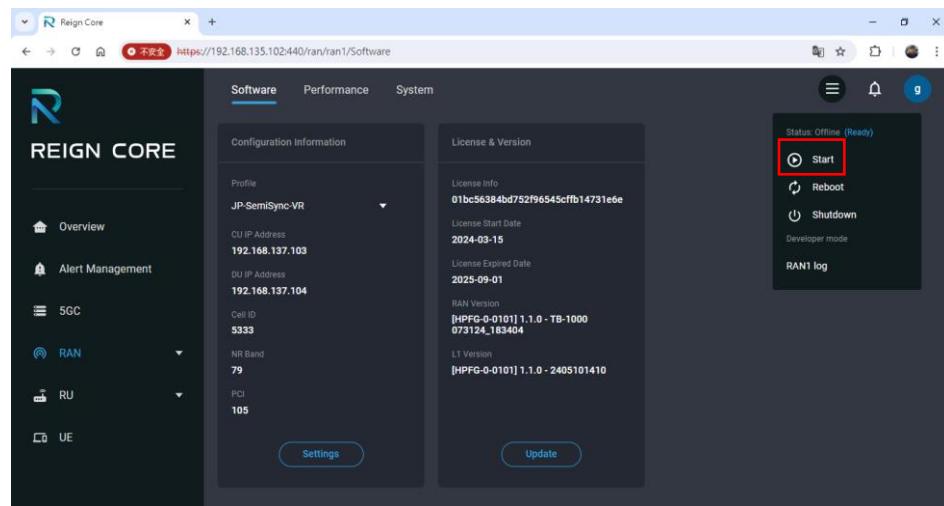
```
$ chmod 777 -R srs_udp_log_reader
```

## 3 Start/Stop RAN Service

Before performing the SRS UDP socket tool, we need to start the service and register the device. Below are the basic steps to start and stop the service. Please ensure the service is started before proceeding with device registration.

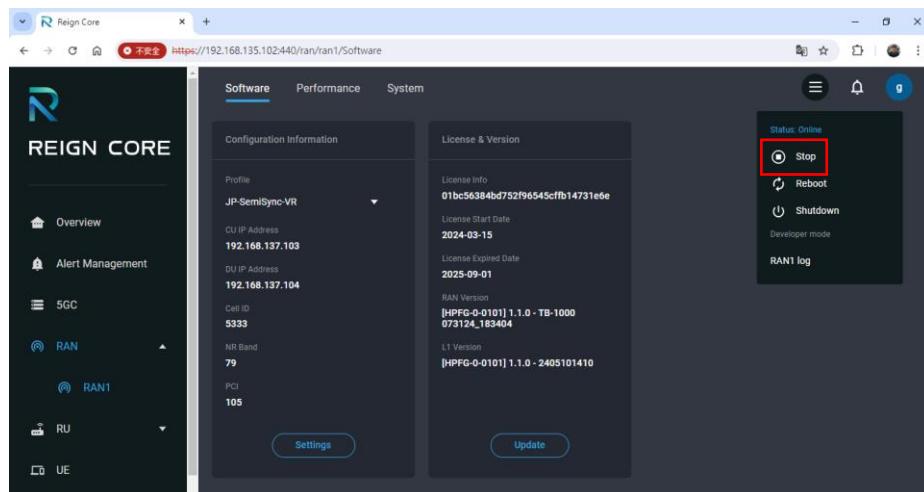
### Start RAN Service

If RAN service is down (status: offline), click the "**Start**" button to start the RAN service. Please note that starting RAN service will take approximately 100 seconds to complete.



### Stop RAN Service

If RAN service is up (Status: online), click the "**Stop**" button to stop the RAN service. Please note that stopping RAN service will take approximately 30 seconds to complete.



## 4 Run SRS UDP Socket Tool

**srs\_udp\_log\_reader** is a tool to instruct you on how to extract the SRS and UTDoA values. After starting the service and registering the device on the network, follow the steps below to execute the SRS UDP socket tool, **srs\_udp\_log\_reader**. The sample code of **srs\_udp\_log\_reader** is located at `/home/htc_l1app/HTC_L1APP/linux/srs_udp_log_reader/main.c`.

### 4.1 xAPP Inside the BBU Server

- a. Create a new SSH session
- b. Change directory to SRS UDP socket tool folder on BBU server  
`$ cd /home/htc_l1app/HTC_L1APP/linux/srs_udp_log_reader`
- c. Execute **srs\_udp\_log\_reader**  
`$ ./srs_udp_log_reader`

You can view the content on the console, as well as in the file "**srs\_udp\_log.log**", which is created in the same folder after executing **srs\_udp\_log\_reader**.

- d. If you want to terminate the **srs\_udp\_log\_reader**, press Ctrl-C to quit.

### 4.2 xAPP Outside the BBU Server

- a. Create a new SSH session
- b. Change directory to SRS UDP socket tool folder on your server  
`$ cd <your path>/srs_udp_log_reader`
- c. Execute **srs\_udp\_log\_reader**  
`$ ./srs_udp_log_reader`

You can view the content on the console, as well as in the file "**srs\_udp\_log.log**", which is created in the same folder after executing **srs\_udp\_log\_reader**.

- d. If you want to terminate the **srs\_udp\_log\_reader**, press Ctrl-C to quit.

## 5 Definition of srs\_udp\_log.h

The Application Programming Interfaces (APIs) for the UDP socket tool are defined in /home/htc\_l1app/HTC\_L1APP/linux/srs\_udp\_log\_reader/srs\_udp\_log.h. This section will explain the parameters used in the SRS UDP socket tool.

```
#define SRS_UDP_LOG_MAX_PORTS      4
#define SRS_UDP_LOG_MAX_ANTS        32
#define SRS_UDP_LOG_MAX_PRB        273
#define SRS_UDP_LOG_IQ              2

typedef struct
{
    uint32_t nBufferIdx;
    struct timeval sTimeStamp;
    uint16_t nCarrierIdx;
    uint16_t nRxFrameNum;
    uint16_t nSlotNum;
    uint16_t nUEId;
    uint16_t nRNTI;
    int16_t nPuschTimeOffset[SRS_UDP_LOG_MAX_ANTS];
    int16_t nPuschEstSnr[SRS_UDP_LOG_MAX_ANTS];
    uint16_t nNrOfPort;
    uint16_t nNrOfRxAnt;
    uint16_t nNrOfRbs;
    int16_t nChanEst[SRS_UDP_LOG_MAX_PORTS][SRS_UDP_LOG_MAX_ANTS][SRS_UDP_LOG_MAX_PRB][SRS_UDP_LOG_IQ];
} srs_udp_log_buf_t;
```

### 5.1 Structure Definitions

#### srs\_udp\_log\_buf\_t

Field	Type	Description
SRS_UDP_LOG_MAX_PORTS		the maximum number of SRS ports
SRS_UDP_LOG_MAX_ANTS		the maximum number of all RU antennas
SRS_UDP_LOG_MAX_PRB		the maximum number of PRB
SRS_UDP_LOG_IQ		the number of IQ type
nBufferIdx	uint32_t	SRS UDP socket log buffer index, value: 0~255
sTimeStamp	timeval struct	Log timestamp
nCarrierIdx	uint16_t	Serving Cell index
nRxFrameNum	uint16_t	System frame number
nSlotNum	uint16_t	Slot number
nUEId	uint16_t	UE index in current slot
nRNTI	uint16_t	The RNTI used for identifying the UE when receiving the PDU
nPuschTimeOffset[SRS_UDP_LOG_MAX_ANTS]	int16_t	Unit of To = 1 /(4096 x 30KHz) = 8.318 ns The Timing Offset is equivalent to UTDoA and is calculated from PUSCH for this user. Each RU has 4 antennas, and with 8 RUs, there are a total of 32

		antennas, resulting in 32 To values
nPuschEstSnr[SRS_UDP_LOG_MAX_ANTS]	int16_t	The Signal-to-Noise Ratio (SNR) is calculated from PUSCH for this user. Each RU has 4 antennas, and with 8 RUs, there are a total of 32 antennas, resulting in 32 SNR values
nNrOfPort	uint16_t	Number of SRS ports for this user
nNrOfRxAnt	uint16_t	Number of Rx antennas for this user
nNrOfRbs	uint16_t	Number of RBs based on numerology and bandwidth for this user
nChanEst[SRS_UDP_LOG_MAX_PORTS] [SRS_UDP_LOG_MAX_ANTS] [SRS_UDP_LOG_MAX_PRB] [SRS_UDP_LOG_IQ];	int16_t	This is SRS Channel Estimation matrix. The nChanEst is an intel 16s13 format, so it needs to be divided by 8192. We did not divide it by 8192. After printing it out, you will need to process it yourself. EX: (450,-497i) and the real IQ value is (450/8192,-497/8192i)=(0.055,-0.06i)