

Command Line Interface and Program Option Reference



PicoScenes Command Line Interface

As shown in the [scenarios](#), multiple command options can be written in a single command string, which is surrounded by quotation marks. This section uses the following example commands to describe how PicoScenes parses the program options.

```
PicoScenes "-d debug;
            // the following commands are for each NIC.
            -i 4 --freq 2412e6 --rate 20e6 --mode responder --rxcm 3 --cbw 40 --sts 2 --
txcm 5 -ess 1 --txpower 15 --coding ldpc;
            -i 3 --freq 2412e6 --rate 20e6 --mode initiator --repeat 100 --delay 5000 --
cf 2412e6:5e6:2484e6 --sf 20e6:5e6:40e6 --cbw 20 --sts 2 --mcs 0 --gi 400 --txcm 3 --ack-mcs
3 --ack-type header;
            -q"
```

The first step of command parsing is to segment the long string into several ; ended command sentences . The second command sentence starts with // or # , it is recognized as a line of comment and is skipped. The remaining command sentences are:

1. -d debug
2. -i 4 --freq 2412e6 --rate 20e6 --mode responder --rxcm 3 --cbw 40 --sts 2 --txcm 5 -ess 1 --txpower 15 --coding ldpc ;
3. -i 3 --freq 2412e6 --rate 20e6 --mode initiator --repeat 100 --delay 5000 --cf 2412e6:5e6:2484e6 --sf 20e6:5e6:40e6 --cbw 20 --sts 2 --mcs 0 --gi 400 --txcm 3 --ack-mcs 3 --ack-type header ;
4. -q

Each command sentence is composed of one or multiple program options . Each program option usually starts with -- . For some frequently used options, we provide shortcuts, like -i is for --interface , -q is for --quit and -d is for --log-display-level . Each program option consists of the option name and zero or more parameters, for example --no-hp doesn't require a parameter, and users may provide multiple parameters for --tx-channel , like --tx-channel 0,1,2,3 .

PicoScenes parses and executes the command sentence in input order. For the program options within one command sentence, PicoScenes invokes **four levels of program option parsers to parse them**; therefore, the order of the program options does not matter. The four levels of parsers, in their hierarchical order, are Platform Startup Options, Platform Options, Frontend Level Options, and Per-Plugin Level Options. They are detailed in the following Program Options Hierarchy. Platform Startup Options and Platform Options are frontend-irrelevant, and the other two levels of parsers are frontend-related.

The following process simulates the real parsing process of the above command sentences.

1. For the first command sentence `-d debug` :

1. Platform Startup Option parser recognizes the `-d debug` option, and sets the debug message display level to `debug` .
2. Platform Option parser recognizes nothing.
3. As no frontend is specified, Frontend Option parser and Per-Plugin Option parsers are skipped.

2. For the second command sentence `-i 4 --freq 2412e6 --rate 20e6 --mode responder --rxcm 3 --cbw 40 --sts 2 --txcm 5 -ess 1 --txpower 15 --coding ldpc` :

1. Platform Startup Option parser recognizes nothing.
2. Platform Option parser recognizes `-i 3` , indicating that the rest of the program options within this command sentence are all for NIC `3` .
3. Frontend Option parser recognizes 5 hardware-tuning options: `--freq 2412e6` , `--rate 20e6` , `--rxcm 3` , `--txcm 5` , and `--txpower 16` . All these controls are for NIC `3` , and Frontend Option parser doesn't perform the hardware tuning but delegates these parameters to the FrontEnd class.
4. Each PicoScenes plugin has the ability to parse command sentences, and PicoScenes enumerates all the active plugin instances of NIC `3` and lets them parse the current command sentence. In the above example, the program option parser of EchoProbe plugin recognizes 5 options, namely, `--mode responder` , `--cbw 40` , `--sts 2` , `--ess 1` and `--coding ldpc` . The `responder` mode of EchoProbe plugin is not a thread-blocking mode, therefore, the parser of EchoProbe plugin will save the parameters and exit.
5. PicoScenes continues to enumerate the rest of the plugins (if there are) and lets them parse the same command sentence.
6. All four levels of parsers finish their jobs, PicoScenes continues to the next command sentence.

3. For the third command sentence `-i 3 --freq 2412e6 --rate 20e6 --mode initiator --repeat 100 --delay 5000 --cf 2412e6:5e6:2484e6 --sf 20e6:5e6:40e6 --cbw 20 --sts 2 --mcs 0 --gi 400 --txcm 3 --ack-mcs 3 --ack-type header` :

1. Platform Startup Option parser recognizes nothing.

2. Platform Option parser recognizes `-i 4` , indicating this the rest of the program options are all for NIC `4` .
 3. Frontend Option parser recognizes 3 program options `--freq 2412e6` , `--rate 20e6` , `--txcm 3` . All these controls are for NIC `4` , and Frontend Option parser doesn't perform the hardware tuning but delegates these parameters to the FrontEnd class.
 4. PicoScenes enumerates all the active plugin instances of NIC `4` and lets them parse the current command sentence. The program option parser of EchoProbe plugin recognizes 11 options, namely, `-mode initiator` , `--repeat 100` , `--delay 5000` , `--cf 2412e6:5e6:2484e6` , `--sf 20e6:5e6:40e6` , `--cbw 20` , `--sts 2` , `--mcs 0` , `--gi 400` , `--ack-mcs 3` and `--ack-type header` . The `initiator` mode of EchoProbe plugin is a blocking mode. NIC `4` 's EchoProbe plugin instance will perform the round-trip and spectrum-scanning CSI measurement. When the measurement finishes or fails, EchoProbe will exit the blocking state.
 5. PicoScenes continues to enumerate the rest of the plugins (if there are) and lets them parse the same command sentence.
 6. All four levels of parsers finish their jobs, PicoScenes continues to the next command sentence.
4. For the fourth command sentence `-q` :
1. Platform Startup Option parser recognizes nothing.
 2. Platform Option parser recognizes `-q` and triggers PicoScenes shutdown sequence.

Program Options Hierarchy

Various PicoScenes program options are organized in a hierarchical structure as listed below:

- **Per-Plugin level Options (Top)**
 - Each PicoScenes plugin can have its own program options. For example, the EchoProbe plugin has a large set of options controlling the packet injection and round-trip measurement.
- **Frontend Level Options**
 - PicoScenes provides two independent sets of options for QCA9300/IWL5300 and SDR frontends, respectively. Users specify different options for different Wi-Fi NICs or USRPs.
- **Platform Options**
 - These are a few global options valid *after* the launch of the PicoScenes platform.
- **Platform Startup Options (Bottom)**
 - These are a few global options valid *before* the launch of the PicoScenes platform.

Tips

You can also look up the **complete** program options by running the command `PicoScenes --help` , if you have successfully installed the PicoScenes.

Platform Startup Options (Bottom)

- `--plugin-dir <new_plugin_dir>`
 - Description: Change the plugin search directory to your specified directory, e.g. `--plugin-dir /home/YOUR_HOME/PicoScenes-PDK` . If not specified, PicoScenes will by default search for plugins in `/usr/local/PicoScenes/plugins`.
 - Default: `/usr/local/PicoScenes/plugins`
 - Value Range: N/A
 - Notes: No
 - Example: `--plugin-dir /home/YOUR_NAME/PicoScenes-PDK`
- `-d [--display-level] <log_display_level>`
 - Description: Specify the log message display level.
 - Default: `info`
 - Value Range: `[vv , verbose , debug , detail , trace , info , warning , error , mute]`
 - Notes: `vv` is the 'most verbose' mode and `mute` silences all log display.
 - Example: `-d trace`
- `--no-hp`
 - Description: Disable process priority escalation. `PicoScenes` will by default try to escalate its process priority to improve the performance, however, it *may* fail due to insufficient privilege. You may specify `--no-hp` to disable the priority escalation.
 - Default: N/A
 - Value Range: N/A
 - Notes: N/A
 - Example: N/A

Platform Options

- `-i [--interface] arg`

- Description: The ID of the target device/interface. This value MUST be provided to validate the Frontend Options.
 - For QCA9300/IWL5300 NIC, PhyId, DvId, MonId and PhyPath are all acceptable. (What are these IDs? You may refer to [Device Naming](#) for help.)
 - For a single network-connected N210/X310 USRP, the name should be `usrp<ip address of the USRP>` , e.g., `usrp192.168.10.2` .
 - For a single PCI-E cable-connected X310 USRP, the name should be `usrp<Resource Id of the X310>` , e.g., `usrpRI00` .
 - To combine multiple network-connected (or MIMO cable connected) N210s/X310s, the name should be `usrp<ip address of the USRP1,ip address of the USRP2,ip address of the USRP3...>` , e.g., `usrp192.168.40.2,192.168.41.2` .
 - Default: N/A
 - Value Range: N/A
 - Notes:
 - Note 1: For USRP, You can lookup the IP address or Resource ID via the UHD facility `uhd_find_devices` .
 - Note 2: For network connected USRPs, you MUST pay special attention to check the matching of IP addresses between the IP address of USRP and the IP address of your NIC. `uhd_find_devices` may find USRP devices even under mismatched address spaces, however, PicoScenes cannot initialize the USRP device in the mismatched situation.
 - Example: `-i usrp192.168.10.2 , -i 3 , -i wlp3s0 , -i phy0 , -i phy0mon`
- `-h [--help]`
 - Description: Show help information.
 - Default: N/A
 - Value Range: N/A
 - Notes: N/A
 - Example: `-h`

Frontend Level Options

PicoScenes provides different options for QCA9300/IWL5300 NICs and USRPs.

QCA9300/IWL5300 NIC options

- `--freq arg`
 - Description: Specify the carrier frequency for both the QCA9300 and IWL5300. This option supports the scientific notation like 2412e6 or 2.412e9.

- Default: The default value is the current working carrier frequency.
 - Value Range:
 - For QCA9300, its frequency synthesizer supports ranges of [2.2-2.9] GHz and [4.4 - 6.1] GHz in the 2.4 and 5 GHz bands, respectively. You can specify any carrier frequency within the ranges.
 - For IWL5300, you can only specify the standard channel frequencies, e.g., 2412e6, 2432e6, 5815e6, etc.
 - Notes:
 - We have observed the decline of Tx/Rx performance caused by the cross-band tuning, e.g., 2412e6->5200e6. We recommend using `array_prepare_for_picoscenes` to perform the cross-band tuning.
 - When operating in `HT40+/-` channel modes, this option, which always refers to the real carrier frequency, is not equal to the center frequency of `HT40+/-`'s primary channel, e.g., if you want to communicate with a `5200 HT40-` channel, you should tune your carrier frequency to 5190e6 or 5200e6 with 40 or 20 MHz channel bandwidth (CBW), respectively.
 - Example: `--freq 5200e6`
- `--rate arg`
 - Description: Specify the baseband sampling rate (or bandwidth) for both the QCA9300 and IWL5300. This option supports the scientific notation like `20e6` or `25e6`.
 - Default: The default value is the current working baseband bandwidth.
 - Value Range:
 - For QCA9300, the available rates under `HT20` channel mode are [2.5, 5, 7.5, 10, 12.5, 15, 17.5, 20, 25, 30, 35, 40] MHz; for `HT40+/-` channel modes the supported rates are [5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80] MHz.
 - For IWL5300, the driver does NOT support bandwidth arbitrary tuning, so this option only supports 20 or 40 MHz.
 - Notes: When `HT20` mode communicates with `HT40+/-` modes with a non-standard bandwidth, you should tune the carrier frequency of the `HT20` side to the correct value. For example, with 20 MHz real bandwidth, `HT40-` channel mode at the 5190 MHz can ONLY communicate with a `HT20` mode with 5195 MHz carrier frequency.
 - Example: `--rate 20e6`
 - `--txcm arg`
 - Description: Specify the transmit chain(s) for the QCA9300 and IWL5300 NICs. The mask is in 3-bit format, i.e., 1/2/4 for the 1st/2nd/3rd chain, 3 for both the 1st and 2nd chains and 7 for all three chains.
 - Default: This value is 7 by default and is persistent until the next NIC reset.
 - Value Range: [1, 2, 3, 4, 7]
 - Notes:

- When the number of the transmit chains(s), $N_{\{tx\}}$, is smaller than the number of transmit spatial-time streams, $N_{\{sts\}}$, the transmission is invalid.
 - Values 5 and 6 are not valid for both QCA9300 and IWL5300.
- Example: `--txcm 1`
- `--rxcm arg`
 - Description: Specify the receive chain(s) for the QCA9300 and IWL5300 NICs. This option has the identical format as `--txcm` option.
 - Default: This value is 7 by default and is persistent until the next NIC reset.
 - Value Range: 1, 2, 3, 4, 7
 - Notes:
 - When the number of the receive chains(s), $N_{\{rx\}}$, is smaller than $N_{\{sts\}}$ of the transmitted packets, the receiver cannot decode the frame.
 - Values 5 and 6 are not valid for both QCA9300 and IWL5300.
 - Example: `--rxcm 1`
- `--txpower arg`
 - Description: Specify the transmit power (Tx power) in dBm for both the QCA9300 and IWL5300.
 - Default: 20
 - Value Range: 0 dBm ~ 30 dBm
 - Notes: This value is 20 by default and is persistent until the next NIC reset.
 - Example: `--txpower 15`
- `-p [--cf-tuning-policy] arg`
 - Description: Specify the tuning policy for QCA9300's carrier frequency. You can specify one of the three policies: `chansel`, `fastcc` and `reset`.
 - Default: `fastcc`
 - Value Range: `chansel`, `fastcc` and `reset`
 - Notes:
 - `chansel` refers to the direct tuning of the RF frequency synthesizer via hardware registers. Since this policy tunes ONLY the synthesizer and bypasses many other settings, this is the fastest but also the least reliable policy.
 - `fastcc` refers to the FAST Channel Change protocol in ath9k driver. This is the default policy in both the ath9k driver and PicoScenes. In ath9k driver, `fastcc` handles the non-crossband channel change scenarios.
 - `reset` refers to the longer and more complete channel change protocol in ath9k driver, which includes hardware reset. In ath9k driver `reset` handles the cross band channel change.
 - Example: `-p chansel`

USRP frontend options

- `--freq arg`
 - Description: Specify the carrier frequency for SDR frontend. This option supports the scientific notation like 2412e6 or 2.412e9.
 - Default: This option has NO default value and is not persistent. You should specify it every time.
 - Value Range: Hardware decides the range.
 - Notes:
 - The value range is based on your hardware. For example, UBX-40/160 daughterboard supports a range of 10-6000MHz.
 - This option sets the same carrier frequency for both the Tx and Rx chains, though the hardware supports setting different frequencies for them.
 - For multi-channel configurations (both channels of X310, or multiple USRPs synchronized by the MIMO cable or external clock source), this option will set the same frequency for all channels.
 - Example: `--freq 5200e6`
- `--rate arg`
 - Description: Specify the baseband sampling rate (or bandwidth) for SDR frontend. This option supports the scientific notation like 25e6 or 40e6.
 - Default: This option has NO default value and is not persistent. You should specify it every time.
 - Value Range: N/A
 - Notes:
 - The value range is based on your hardware. For example, X310 motherboard supports a maximum sampling rate of 200 MHz.
 - This option sets the same sampling rate for both the Tx and Rx chains, though the hardware supports setting different sampling rates for them.
 - Different hardware has different tuning granularity. For example, you can only specify 200/INT_N MHz sampling rate where INT_N is a positive integer.
 - Example: `--rate 20e6`
- `--tx-resample-ratio arg`
 - Description: Specify the Tx resampling ratio. This is a software-based Tx resampling mechanism to enable arbitrary channel bandwidth. For example, X310 cannot tune the baseband sampling rate to 80 MHz. To overcome this issue, we can tune the hardware to 100 MHz (by `--rate 100e6`) and then resample the Tx signal by 0.8 (by `--tx-resample-ratio 0.8`).
 - Default: 1.0

- Value Range: 0 ~ 1.0
 - Notes:
 - This option is implemented by zero-order interpolation on the software side, i.e., generate the signal by 80 MHz then interpolate the signal to 100 MHz.
 - This interpolation is implemented by software, therefore the performance decline should be expected.
 - Example: `--tx-resample-ratio 0.8`
- `--rx-resample-ratio arg`
 - Description: Specify the Rx resampling ratio. This is a software-based Rx resampling mechanism to enable arbitrary channel bandwidth. For example, X310 cannot tune the baseband sampling rate to 80 MHz. To overcome this issue, we can tune the hardware to 100 MHz (by `--rate 100e6`) and then resample the Rx signal by 0.8 (by `--rx-resample-ratio 0.8`).
 - Default: 1.0
 - Value Range: 0 ~ 1.0
 - Notes:
 - This option is implemented by uniform signal dropping on the software side, i.e., receive the signal by 100 MHz rate then decimate the signal to 80 MHz.
 - This resample is implemented by software, therefore the performance decline should be expected.
 - Example: `--rx-resample-ratio 1.0`
 - `--clock-source arg`
 - Description: Specify the clock and time source for SDR frontend.
 - Default: `internal`
 - Value Range: `internal`, `external`, `mimo`
 - Notes: You can specify `external` for G-Octoclock based clock source or `mimo` for N210 MIMO-cable based clock source sharing.
 - Example: `--clock-source external`
 - `--cfo arg`
 - Description: Specify the carrier frequency offset for Tx baseband. This option supports the scientific notation like `1e3` (1000 Hz cfo). This option is implemented by Wi-Fi baseband software, therefore the performance decline should be expected.
 - Default: 0.0
 - Value Range: N/A
 - Notes: N/A
 - Example: `--cfo 1e3`
 - `--sfo arg`

- Description: Specify the sampling rate offset for Tx baseband. This option supports the scientific notation like 1e3 (1000 Hz sfo). This option is implemented by Wi-Fi baseband software, therefore the performance decline should be expected.
- Default: 0.0
- Value Range: N/A
- Notes: N/A
- Example: `--sfo 1e3`
- `--master-clock-rate arg`
 - Description: Specify the master clock rate of USRP. For Wi-Fi communication
 - Default: For N210 and X310, the default value is 100e6 and 200e6 respectively.
 - Value Range: N/A
 - Notes: N/A
 - Example: `--master-clock-rate 100e6`
- `--tx-channel arg`
 - Description: Specify the Tx channel(s) for SDR frontend. The default value is 0, which means 0-th channel. Multiple channel numbers are separated by a comma , . In multi-channel configurations, taking two combined X310s for example, you can specify 0,1,2,3 to use all 4 channels for Tx. You can also skip some of them, such as 0,2,3 which specify the 0-th, 1st and 3rd antenna for Tx.
 - Default: 0
 - Value Range: N/A
 - Notes:
 - The order does not matter. 0,2,3 is equal to 3,2,0 .
 - The physical mapping between the channel number and antenna is ordered. For example, assuming that we combine two X310s together with `-i usrp192.168.40.2,192.168.41.2` , 0-th and 1st antennas correspond to the left and right daughterboards of the X310 with IP address 192.168.40.2; and 2nd and 3rd antennas correspond to the left and right daughterboards of the X310 with IP address 192.168.41.2.
 - Example: `--tx-channel 0,1`
- `--rx-channel arg`
 - Description: Specify the Tx channel(s) for SDR frontend. The default value is 0, which means 0-th channel. Multiple channel numbers are separated by a comma , . This option has the identical format as `--tx-channel` .
 - Default: 0
 - Value Range: N/A
 - Notes: N/A
 - Example: `--rx-channel 0,1`

- `--rx-pipe arg --rx-pipe-cfg-file arg`
 - see [MultiPipeLine Config](#)
- `--rx-cbw arg`
 - Description: Specify the Channel Bandwidth (CBW) for Rx baseband. You can specify 20 , 40 , 80 or 160 , which corresponds to 20/40/80/160MHz CBW for Rx baseband.
 - Default: 20
 - Value Range: 20 , 40 , 80 , 160
 - Notes: In order to receive and correctly decode the packet transmitted in HT20/HT40/VHT80/VHT160 formats, you must specify Rx CBW to 20/40/80/160, respectively.
 - Example: `--rx-cbw 40`
- `--rx-ant arg`
 - Description: Specify which RX antenna to use.
 - Default: RX2
 - Value Range: TX/RX , RX2
 - Notes: For USRP UBX/CBX/SBX daughterboard, TX/RX or RX2
 - Example: `--rx-ant TX/RX`
- `--txpower arg`
 - Description: Tx gain.
 - Default: N/A
 - Value Range: 0 ~ 38 dB
 - Notes: N/A
 - Example: `--txpower 20`
- `--rx-gain arg`
 - Description: Rx gain.
 - Default: N/A
 - Value Range: 0 ~ 38 dB
 - Notes: N/A
 - Example: `--rx-gain 20`
- `--filter-bw arg`
 - Description: Baseband filter bandwidth (unit in Hz, scientific notation supported)
 - Default: N/A
 - Value Range: N/A
 - Notes: N/A

- Example: N/A
- `--tx-to-file arg`
 - Description: Supply a file name (without extension, just the name), `PicoScenes` will save all the Tx signals to a file. The signals will be saved to a `.bbsignals` file with the specified file name.
 - Default: N/A
 - Value Range: N/A
 - Notes: N/A
 - Example: `tx-to-file demo`
- `--tx-from-file arg`
 - Description: Supply a file name (without extension, just the name), `PicoScenes` will replay the signal saved in the `.bbsignals` file as if the signal is generated from the baseband.
 - Default: N/A
 - Value Range: N/A
 - Notes: N/A
 - Example: `--tx-from-file demo`
- `--tx-from-file-delay arg`
 - Description: The delay (in ms) before Tx signal replay.
 - Default: N/A
 - Value Range: N/A
 - Notes: N/A
 - Example: `--tx-from-file-delay 1000`
- `--rx-to-file arg`
 - Description: Dump baseband signals received from SDR device to a `.bbsignals` file with the specified file name. This is actually the Rx signal recorder.
 - Default: N/A
 - Value Range: N/A
 - Notes: When `rx-to-file` is ON, the received signal will NOT be sent to baseband for decoding.
 - Example: `--rx-to-file demo`
- `--rx-from-file arg`
 - Description: Replay baseband signals saved in the `.bbsignals` file as if the signal is received from the RF frontend. This is actually the Rx signal replay.
 - Default: N/A
 - Value Range: N/A

- Notes: The Rx signal replay keeps the same pace with the Rx baseband, therefore there will be no signal dropping.
- Example: `--rx-from-file demo`
- `--rx-sensitivity arg`
 - Description: Specify the lowest power level (specified in dB) above the rx noise floor to trigger packet detection.
 - Default: 5
 - Value Range: N/A
 - Notes: N/A
 - Example: `--rx-sensitivity 10`
- `--rx-cp-offset arg`
 - Description: Specify at which position of Cyclic Prefix is regarded as the start of OFDM signal (pre-advancement)
 - Default: 0.75
 - Value Range: 0 ~ 1
 - Notes: N/A
 - Example: `--rx-cp-offset 0.5`
- `--tx-iq-mismatch arg`
 - Description: Specify Tx I/Q mismatch, for example `1.5 15` means 1.5dB Tx I/Q ratio and 15 degree of Tx I/Q crosstalk
 - Default: N/A
 - Value Range: N/A
 - Notes: N/A
 - Example: `tx-iq-mismatch "1.5 15"`
- `--rx-iq-mismatch arg`
 - Description: Specify Rx I/Q mismatch, for example `1.5 15` means 1.5dB Rx I/Q ratio and 15 degree of Rx I/Q crosstalk
 - Default: N/A
 - Value Range: N/A
 - Notes: N/A
 - Example: `rx-iq-mismatch "1.5 15"`
- `--disable-1ant-tx-4-extra-sounding`
 - Description: Enable a special HT-LTF demodulation mode when signal is received from a transmitter with only 1 TX antenna.

- Default: N/A
- Value Range: N/A
- Notes: N/A
- Example: `--disable-1ant-tx-4-extra-sounding`
- `--enable-loopback`
 - Description: Enable USRP Rx loopback signal from Tx.
 - Default: N/A
 - Value Range: N/A
 - Notes: N/A
 - Example: `--enable-loopback`
- `--enable-hw-acc`
 - Description: Enable or disable hardware acceleration for Wi-Fi packet detection (enabling requires our special firmware, false as default).
 - Default: N/A
 - Value Range: N/A
 - Notes: N/A
 - Example: `--enable-hw-acc`

Per-Plugin Level Options (Top)

EchoProbe Options

- `--mode arg`
 - Description: EchoProbe working mode.
 - Default: N/A
 - Value Range: `injector` , `logger` , `initiator` , `responder`
 - Notes: N/A
 - Example: `--mode injector`

EchoProbe initiator options

- `--target-mac-address`
 - Description: MAC address of the injection target [magic Intel `00:16:ea:12:34:56` is default].
 - Default: N/A
 - Value Range: N/A

- Notes: N/A
- Example: N/A
- --5300
 - Description: Both Destination and Source MAC addresses are set to [magic Intel 00:16:ea:12:34:56].
 - Default: N/A
 - Value Range: N/A
 - Notes: N/A
 - Example: N/A
- --cf
 - Description: MATLAB-style specification for carrier frequency scan range, format begin:step:end .
 - Default: N/A
 - Value Range: N/A
 - Notes: N/A
 - Example: --cf 5200e6:20e6:5800e6
- --sf
 - Description: MATLAB-style specification for baseband sampling frequency multiplier scan range, format begin:step:end .
 - Default: N/A
 - Value Range: N/A
 - Notes: N/A
 - Example: --sf 10e6:5e6:20e6
- --repeat
 - Description: The injection number per cf/bw combination.
 - Default: 100
 - Value Range: N/A
 - Notes: N/A
 - Example: --repeat 1e4
- --delay
 - Description: The delay between successive injections(unit in us).
 - Default: 5e5
 - Value
 - Example: --repeat 1e4

- `--delay`
 - Description: The delay between successive injections(unit in us).
 - Default: 5e5
 - Value Range: N/A
 - Notes: N/A
 - Example: `--delay 5e3`
- `--delayed-start`
 - Description: A one-time delay before injection(unit in us)
 - Default: 0
 - Value Range: N/A
 - Notes: N/A
 - Example: `--delayed-start 5e5`
- `--format`
 - Description: 802.11 frame format.
 - Default: HT
 - Value Range: nonHT , HT , VHT , HESU
 - Notes: N/A
 - Example: `--format VHT`
- `--cbw`
 - Description: Channel Bandwidth (CBW) for injection(unit in MHz).
 - Default: 20
 - Value Range: 20 , 40 , 80 , 160
 - Notes: N/A
 - Example: `--cbw 40`
- `--mcs`
 - Description: The MCS index for one single spatial stream.
 - Default: 0
 - Value Range: 0 ~ 11
 - Notes: N/A
 - Example: `--mcs 4`
- `--sts`
 - Description: Number of spatial time stream (STS).
 - Default: 1

- Value Range: 1 ~ 4
- Notes: N/A
- Example: `--sts 2`
- `--ess`
 - Description: Number of Extension Spatial Stream for TX.
 - Default: 0
 - Value Range: 0 ~ 3
 - Notes: N/A
 - Example: `--ess 2`
- `--gi`
 - Description: Guarding Interval.
 - Default: 800
 - Value Range: 400 , 800 , 1600 , 3200
 - Notes: N/A
 - Example: `--gi 1600`
- `--coding`
 - Description: Code scheme.
 - Default: BCC
 - Value Range: LDPC , BCC
 - Notes: N/A
 - Example: `--coding LDPC`
- `--injector-content`
 - Description: Content type for injector mode.
 - Default: full
 - Value Range: full , header , ndp
 - Notes: N/A
 - Example: `--injector-content header`
- `--ifs`
 - Description: Inter-Frame Spacing in seconds.
 - Default: 20e-6
 - Value Range: N/A
 - Notes: N/A
 - Example: `--ifs 10e-6`

Echo responder options

- `--ack-type`
 - Description: EchoProbe reply strategy.
 - Default: full
 - Value Range: full , csi , extra , header
 - Notes: N/A
 - Example: `--ack-type csi`
- `--ack-mcs`
 - Description: MCS value (for one single spatial stream) for ack packets, unspecified as default.
 - Default: N/A
 - Value Range: 0 ~ 11
 - Notes: N/A
 - Example: `--ack-mcs 4`
- `--ack-sts`
 - Description: The number of spatial time stream (STS) for ack packets, unspecified as default.
 - Default: N/A
 - Value Range: 0 ~ 23
 - Notes: N/A
 - Example: `--ack-sts 3`
- `--ack-cbw`
 - Description: Bandwidth for ack packets (unit in MHz), unspecified as default.
 - Default: N/A
 - Value Range: 20 , 40 , 80 , 160
 - Notes: N/A
 - Example: `--ack-cbw 40`
- `--ack-gi`
 - Description: Guarding-interval for ack packets, unspecified as default.
 - Default: N/A
 - Value Range: 400 , 800 , 1600 , 3200
 - Notes: N/A

- Example: `--ack-gi 800`