

# How to measure FA .NET Core performance using the roundtrip latency benchmark samples

For best results, two machines should be used for the test, one as the Sender Host (Client), the other as the Receiver Host (Server), OpenOnload could be used in case of Linux hosts.

FIX Antenna .NET Core performance can be measured by the execution of `runClientRoundTripLatencyBM` and `runServerRoundTripLatencyBM` scripts contained in the package (`\Examples\Latency\Sender` and `\Examples\Latency\Server`).

Configuration file `engine.properties` can be found in the application folder (Sender or Server).

You can reproduce the measurement on your hardware using the following instruction:

1. Open `Benchmarks/Latency/Server` or `Benchmarks/Latency/Sender` folder.
2. On the Receiver Host execute `runServerRoundTripLatencyBM` script
3. On the Sender Host open the `runClientRoundTripLatencyBM` script and change the IP address from the 127.0.0.1 to the IP of the Receiver Host, run the script
4. If a Solarflare Network interface controller is installed on the machine it is possible to activate the Solarflare OpenOnload technique. To achieve this you need to execute the scripts as follows:  
`onload --profile=latency ./runServerRoundTripLatencyBM` on the Receiver Host  
`onload --profile=latency ./runClientRoundTripLatencyBM` on the Sender Host
5. After the test is completed, the `Latency-{timestamp}.hgrm` file with the percentiles data will be created on the Sender Host in the `Results` folder

User can run these samples on Windows (both .NET Core and .NET Framework) using, for example, `runServerRoundTripLatencyBMCore.bat` or `runServerRoundTripLatencyBMNet48.bat` respectively.

## Configuration and running

Command line arguments for **Server** (used in `runServerRoundTripLatencyBM`):

**Server host port** where **host** is host name or IP address of server part and **port** is server's port.

Command line arguments for **Sender** (used in `runClientRoundTripLatencyBM`):

**Sender host port rateLimited rate** where **host** is host name or IP address of server part, **port** is server's port, **rateLimited** (`true/false`) show is messages' rate is limited, and if rate is limited, then **rate** parameter defines messages rate in msgs/sec (up to 1000).

User can configure CPU affinity in the `engine.properties` file. By default affinity is set to CPU 0 for both sending and receiving threads:

```
# This parameter specifies CPU id for the threads of session that send and receive the data from/in socket.  
# Please set correct CPU id for better performance  
# By default used first CPU (id=0)  
cpuAffinity=0
```

Instead of single option '`cpuAffinity`' user can use '`recvCpuAffinity`' and '`sendCpuAffinity`' configuration options to pin sending and receiving threads to different CPU cores.