

Using MEASURE Counters

LightWave Client supports several 'user defined' MEASURE counters which you can use to help analyze performance of your LightWave Client application. Each counter is an array counter in which each element of the array corresponds to an operation within the API that LightWave Client is currently supporting. For example, the nth index of the LWC-CALL-COUNT counter corresponds to the nth index of the API operation array. To display the API operations and their indexes, use the LWCCOM utility.

The available MEASURE counter names and types are shown below.

Counter Name	Type	Description
lwc-call-count	accum, array	The number of API calls.
lwc-connect	qbusy, array	The amount of time spent connecting to the remote API endpoint.
lwc-serialize	qbusy, array	The amount of time spent serializing the API request (converting the structured IPM to JSON format).
lwc-sign	qbusy, array	The amount of time spent signing the API request, when required.
lwc-request	qbusy, array	The amount of time spent sending the request to the remote API endpoint.
lwc-response	qbusy, array	The amount of time spent receiving the response from the remote API endpoint.
lwc-deserialize	qbusy, array	The amount of time spent de-serializing the API response (converting the JSON formatted message to the structured IPM format).
lwc-diag-log	qbusy, array	The amount of time spent generating the diagnostic log.
lwc-msg-log	qbusy, array	The amount of time spent filtering and delivering the message logging events.
lwc-total	qbusy, array	The total amount of time spent processing the request and response. This value may not exactly match the sum of the other busy counters due to some miscellaneous processing not measured by the other counters. Note that the total time does not account for time spent waiting for the remote API service endpoint to respond.



When adding counters to your measurement, be sure to specify arrays that are at least as large as the number of operations (paths/methods) in the API, otherwise some operations will not be measured.

```
> VOLUME <lwc-installation-subvolume>
> MEASCOM

+ add userdef CLIENT

== where ARRAY 10, for example, represents the number of
== operations in the API in use; each one has its own count

+ add counter lwc-call-count, process client, accum, array 10
+ add counter lwc-connect, process client, qbusy, array 10
+ add counter lwc-serialize, process client, qbusy, array 10
+ add counter lwc-sign, process client, qbusy, array 10
+ add counter lwc-request, process client, qbusy, array 10
+ add counter lwc-response, process client, qbusy, array 10
+ add counter lwc-deserialize, process client, qbusy, array 10
+ add counter lwc-diag-log, process client, qbusy, array 10
+ add counter lwc-msg-log, process client, qbusy, array 10
+ add counter lwc-total, process client, qbusy, array 10

+ start measurement LWCMEAS
== run your application
+ stop measurement LWCMEAS

+ list userdef *, rate off
```



These MEASURE counters may differ from values recorded in diagnostic logs. The diagnostic log measures elapsed time whereas the MEASURE counters represent actual work time.