

How to Obtain Time History Data

Preparation

1. Setup the meter to capture Time History data by going to the Setup Manager > Time History tab then check the Enable Time History checkbox and set the period to 1 s and enable L_{AEQ} , L_{ZPEAK} and L_{APeak} by checking the associated checkboxes. Click Close and save the Settings. Click Close again.
2. Start a measurement. The meter will start collecting Time Histories.
3. Now you can, as directed in the SDK, connect directly to the meter over TCP/IP or launch HLD to connect over USB.

Requesting Time Histories from HLD

The following calls will provide data in the form of JSON that comes from the collected Time Histories:

timeHistJson is used to request up to 120 Time History data points at a time.

Params:

group: string – can either be:

1. “time” for current Time History data or
2. “filetime” if a file has been opened and prepared for requests

metric: int - specifies which of the Time History Options to return in thMetrics. Zero based index to the metrics selected in your settings. If you checked 3 settings then 0, 1 and 2 are valid options.

index: int - the index of the first record to retrieve with the request

Returns:

thLeq: array of floats – Leq for each time period interval defined in the settings

thMetrics: array of floats – metric values from request

thFlags: array of flags (ints) – used to determine what the Time History entry represents. If an entry in thFlags is 0 then the same entry in thActions has no meaning. Each thFlags entry may be some combination of the following flags:

```
TIMEHIST_MARKER1 = (0x00000001);  
TIMEHIST_MARKER2 = (0x00000002);  
TIMEHIST_MARKER3 = (0x00000004);  
TIMEHIST_MARKER4 = (0x00000008);  
TIMEHIST_MARKER5 = (0x00000010);  
TIMEHIST_MARKER6 = (0x00000020);  
TIMEHIST_MARKER7 = (0x00000040);  
TIMEHIST_MARKER8 = (0x00000080);  
TIMEHIST_MARKER9 = (0x00000100);  
TIMEHIST_MARKER10 = (0x00000200);  
TIMEHIST_OVERLOAD = (0x00000400);  
TIMEHIST_OBA_OVLD = (0x00000800);  
TIMEHIST_EXCDED = (0x00001000);  
TIMEHIST_T2READY = (0x00002000);  
TIMEHIST_PARTIAL = (0x00002000);
```

```
TIMEHIST_SR = (0x00004000);  
TIMEHIST_BACKERASE = (0x00008000);  
TIMEHIST_MANUAL = (0x00010000);  
TIMEHIST_SESSION_LOG = (0x80000000);
```

thTime: array of ints – each entry is the number of seconds since epoch that is the start of this Time History entry.

thActions: array of ints – if the entry in thFlags == TIMEHIST_SESSION_LOG then the entry in thActions can be split into 2 parts: action and cause.

```
var action = (action & 0x00FF).toString();  
var cause = (action & 0xFF00).toString();
```

action is index into the following table and uses the same table for Session Log entries:

Action	Description
0	Error
1	Stop
2	Run
4	Pause
8	Resume
16	Voice
32	Sound
64	Cal. Check
128	Clear
129	GPS Time Sync
130	Back Erase
131	Mark
132	Cal. Change
133	Preamplifier Disconnected
134	Preamplifier Connected
135	File Avg
136	Comms Watchdog
137	USB
138	Panic
139	Charging Stopped
140	NTP Time Sync
141	Time Adjustment

cause is an index into the following table and is the same as Session Log entries:

Cause	Description
256	KeyPress
257	Event
258	Positive Adjustment

512	IO Command
513	Measurement
514	Negative Adjustment
1024	Timer
1025	Markers
2048	Power
4095	Out of Memory
8192	Pream Connected
16384	Preamp Disconnected
32768	Stable
33024	Comms Watchdog
33280	Comms Watchdog
33536	Other
33792	Internal Fault
34048	Other
34304	Too Hot or Cold
34560	Cover Coltage
34816	Batt Age or Type
35072	Low Input Power

The following two requests simply change the metric from $L_{AEQ} = 0$ to $L_{APeak} = 1$. They both start at index = 0 meaning they start at the first entry of the Time History. Note that the first entry is not a valid L-value and represents the Run. You can use the thFlags for the Time History to know whether each entry is a valid L-value or represent other information.

Request:

`/sdk?func=timeHistJson&group=time&metric=0&index=0`

Note that when metric is 0, thMetrics matches the thLeq.

Results:

```
{  "thLeq": [-1.000000e+12, 2.451547e+01, 2.437527e+01, 2.435137e+01, 2.450543e+01,
2.455994e+01, 2.436919e+01, 2.445354e+01, 2.433140e+01, 2.438669e+01, 2.435610e+01,
2.447337e+01, 2.440771e+01, 2.436856e+01, 2.442876e+01, 2.434426e+01, 2.437016e+01,
2.440012e+01, 4.741119e+01, 8.498555e+01, 8.944057e+01, 4.380427e+01, 2.430421e+01,
2.435870e+01, 2.435785e+01, 2.430629e+01, 2.434739e+01, 2.531319e+01, 8.712630e+01,
4.140762e+01, 2.447169e+01, 2.460907e+01, 2.429511e+01, 2.436376e+01, 2.434558e+01,
2.444400e+01, 2.437815e+01, 2.431917e+01, 2.441491e+01, 2.434878e+01, 2.439108e+01,
2.440505e+01, 2.441379e+01, 2.433165e+01, 2.446185e+01, 2.460327e+01, 2.435218e+01,
2.443194e+01, 2.889804e+01, 2.468577e+01, 2.437490e+01, 2.431110e+01, -4.485347e+02],

  "thMetrics": [-1.000000e+12, 2.451547e+01, 2.437527e+01, 2.435137e+01, 2.450543e+01,
2.455994e+01, 2.436919e+01, 2.445354e+01, 2.433140e+01, 2.438669e+01, 2.435610e+01,
2.447337e+01, 2.440771e+01, 2.436856e+01, 2.442876e+01, 2.434426e+01, 2.437016e+01,
2.440012e+01, 4.741119e+01, 8.498555e+01, 8.944057e+01, 4.380427e+01, 2.430421e+01,
2.435870e+01, 2.435785e+01, 2.430629e+01, 2.434739e+01, 2.531319e+01, 8.712630e+01,
```


6.481577e+01, 6.481577e+01, 1.040818e+02, 1.097761e+02, 6.481577e+01, 6.481577e+01,
6.481577e+01, 6.481577e+01, 6.481577e+01, 6.481577e+01, 6.481577e+01, 1.096076e+02,
6.481577e+01, 6.481577e+01, 6.481577e+01, 6.481577e+01, 6.481577e+01, 6.481577e+01,
6.481577e+01, 6.481577e+01, 6.481577e+01, 6.481577e+01, 6.481577e+01, 6.481577e+01,
6.481577e+01, 6.481577e+01, 6.481577e+01, 6.481577e+01, 6.481577e+01, 6.481577e+01,
6.481577e+01, 6.481577e+01, 6.481577e+01, 6.481577e+01, 6.481577e+01, -1.000000e+12],

"thFlags": [2147483648, 0,
0, 8192, 2147483648],

"thTime": [1508401182, 1508401182, 1508401183, 1508401184, 1508401185, 1508401186,
1508401187, 1508401188, 1508401189, 1508401190, 1508401191, 1508401192, 1508401193,
1508401194, 1508401195, 1508401196, 1508401197, 1508401198, 1508401199, 1508401200,
1508401201, 1508401202, 1508401203, 1508401204, 1508401205, 1508401206, 1508401207,
1508401208, 1508401209, 1508401210, 1508401211, 1508401212, 1508401213, 1508401214,
1508401215, 1508401216, 1508401217, 1508401218, 1508401219, 1508401220, 1508401221,
1508401222, 1508401223, 1508401224, 1508401225, 1508401226, 1508401227, 1508401228,
1508401229, 1508401230, 1508401231, 1508401232, 1508401233],

"thAction": [514, 1065353216, 1072902963, 1077097267, 1080872141, 1083388723,
1085276160, 1087163597, 1089051034, 1090728755, 1091672474, 1092616192, 1093559911,
1094503629, 1095447347, 1096391066, 1097334784, 1098278503, 1099064935, 1099536794,
1100008653, 1100480512, 1100952371, 1101424231, 1101896090, 1102367949, 1102839808,
1103311667, 1103783527, 1104255386, 1104727245, 1105199104, 1105670963, 1106142823,
1106614682, 1107086541, 1107427328, 1107663258, 1107899187, 1108135117, 1108371047,
1108606976, 1108842906, 1109078835, 1109314765, 1109550695, 1109786624, 1110022554,
1110258483, 1110494413, 1110730343, 1110861415, 257],

"Result": "Success: 0 ", "ResultCode": 0, "ResultName": "Success"

}

Review the “*get*” function and appendix G “Examples of Using the *get* Function” in ***Using HttpLD (SDK)*** document. Examples shown are for working with the OBA data on Time History and which parameters can be used with ***get***.