Overview

The TrueSight Meter now supports acting as a StatsD metrics aggregator. It is intended to be an "out of the box" replacement for other aggregators, such as Collectd/statsd, and provides additional features outlined below. By default, the StatsD listener in the meter is disabled. This is to insure we don't cause conflicts with other existing solutions that may be present on the customers host machine.

We currently support four metrics types - Timers, Counters, Gauges, and Sets. We additionally support the Dogstatsd extensions for events and tagging.

Further information regarding the core metrics types can be found at StatsD Metrics.

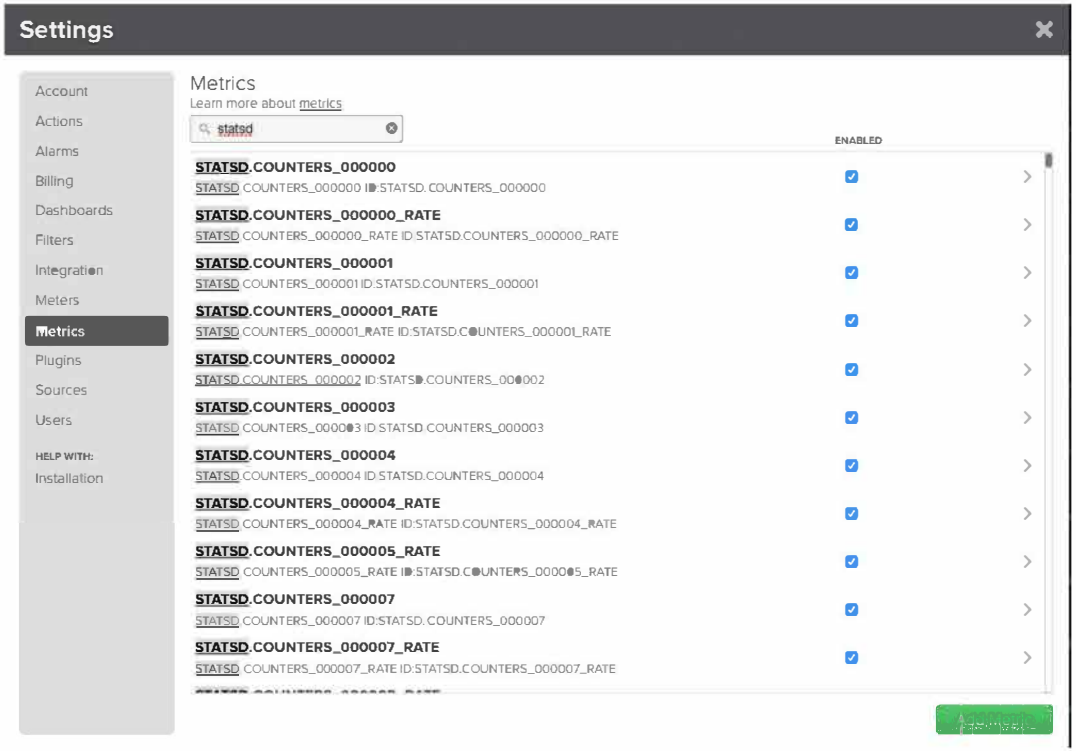
Additional information regarding client support libraries in a variety of languages can be found under the **Client Implementations** section at StatsD client libraries.

Quickstart Guide

To enable StatsD aggregation on the Meter, perform the following actions

- Stop the Meter if it's running
- Edit the meter.conf file, and change "enable" : false to "enable": true in the statsd_sink section.
- Save your changes and exit.
- Start the Meter

The Meter will now listen for StatsD UDP packets on port 8125. Metrics going to the back end will be prefixed with "statsd" in the Metrics UI. Here is a sample, with randomly generated metric names and values.



At this point, the user may want to edit the statsd metrics of interest and change the names to something meaningful to the user, then add the metrics to a custom dashboard. Additionally, the Meter status page contains StatsD configuration information, as well as live counts over the last hour. This can be used to assist in trouble-shooting if necessary. The status page can be access at http://<host>:9193/info

Statsd Service

Setting	Value
Enable	True
Prefix	statsd
Flush Interval Sec	10
UDP Port	8125
TCP Port	0
Counter Prefix	counters
Timer Prefix	timers
Gauge Prefix	gauges
Set Prefix	sets

Statsd Live Counts (Last Hour)

Type	Count
Counters	1633200
Timers	652890
Gauges	980193
Sets	1634303
Events	0
Bad Metrics	0
Bad Events	0
Unknown Metrics	0
Bad Packets	0

Configuration Guide

The basic setup outlined above will get most users started. This section discusses the various other options available via the configuration settings. The StatsD settings are located in the "statsd_sink" section in the meter.conf file. The configuration file is in JSON and may be edited with standard command line tools - eg: vi, nano, emacs, etc.

Setting	Default	Description

	Value	
enable	false	Enables or disables listening for statsd metrics
prefix	"statsd"	The prefix value for each metric
flush_interval_s	10	The interval in seconds between metrics flushes
network.localhost_only	false	Only listen on the localhost address
network.udp_port	8125	The UDP listen port. Setting to 0 disables UDP
network.tcp_port	0	The TCP listen port. Setting to 0 disables TCP
counters_config.extended_counters	false	If enabled, the emitted counters will include summary values from extended_counters_include
counters_config.extended_counters_include	N/A	Options are "count", "mean", "stdev", "sum", "sum_sq", "lower", "upper", "rate"
counters_config.delete_counters	false	Delete counters after flush
counters_config.prefix_counter	"counters"	The name to use for Counters
timers_config.extended_timers	false	If enabled, the emitted timers will include summary values from extended_timers_include
timers_config.extended_timers_include	N/A	Options are "count", "mean", "stdev", "sum", "sum_sq", "lower", "upper", "rate", "median", "sample_rate"
timers_config.prefix_timer	"timers"	The name to use for Timers
timers_config.quantiles	N/A	A comma-separated list of quantiles to calculate for timers if extended_timers is enabled. eg: 0.50, 0.90, 0.99 - will track 50%, 90% and 99% quantiles.
timers_config.delete_timers	false	Delete timers after flush
gauges_config.prefix_gauge	"gauges"	The name to use for Gauges
gauges_config.delete_gauges	false	Delete gauges after flush
sets_config.prefix_set	"sets"	The name to use for Sets
sets_config.delete_sets	false	Delete sets after flush

sets_config.send_zeros	true	Send a 0 metric value if there have no updates since the last flush interval
sets_config.exact_hash_size	64	The size of the internal hash to use for exact matches. Increases in size increase accuracy but at the cost of more memory. In the event the exact hash size is exceeded, an approximation will be made.

Instrumentation Sample

While many clients choose to use one of the myriad of StatsD client libraries available, for simple monitoring, a shell script may suffice.

To illustrate below, we will create a small bash script that pings a service, pulls out the ping time, and sends it to the meter as a timer metric.

Our script - ping.sh

```
#!/bin/bash
while true;
do
    # Extract the ms value from the ping reponse.
    ms=$(ping -c 1 www.google.com | grep time= | awk -F = '{print $4}' | sed 's/ ms//g');
    # Format the ms value in a Statsd timer metric, and push it out to UDP port 8125
    echo "google_ping_time:$ms|ms" | nc -u -w0 127.0.0.1 8125;
   sleep 1
done
```

Enable the StatsD service in the meter following the steps in the QuickStart section. Additionally, set the *timers_config.extended_timers* to true in the meter.conf file.

Start the meter if it's not currently running.

Run the ping.sh script.

Within 10 to 15 seconds, the new metrics will appear in the metrics UI. At this point, they can be given new names if desired, and added existing or new dashboards.