

# Configuring user defined metrics for SNMP

---

Because of the large numbers of metrics available in the 128T platform and because only a fraction of these are useful for a functional Network Management System (NMS) the t128MetricsTable is only populated with a few metrics by default. If required, any 128T metric can be provided in this table via a user configurable JSON file.

## Metrics Table Overview

---

The following are the two value columns of the t128MetricsTable:

```
t128MetricAlias.t128MetricIndex.t128MetricValue  
t128MetricAlias.t128MetricIndex.t128MetricContributors
```

Each row of the metrics table is keyed by two objects, the t128MetricAlias and the t128MetricIndex. The t128MetricAlias is an arbitrary string of up to 64 characters long that represents a description of the metric. When read from the table this alias also includes a series instance suffix, which represents a unique instance of metrics. In this context an instance of metrics represents one set of qualified metrics. For example the default CPU utilization metric has an alias "cpuUtilization" will be returned as "cpuUtilization\_0". In addition to the alias the t128MetricIndex key represents a unique time series of a metric and representing one set of metrics "contributors".

The values contained in each row are the t128MetricValue and t128MetricContributors objects. The t128MetricValue is simply the current value of the metric. To understand the t128MetricContributors object requires a little more understanding of the way 128T metrics are stored and modelled. All metrics in 128T are assigned a "Metric ID" that is modeled in the YANG metrics data-model. This modelled metric ID is typically represented in a path like syntax, for example the CPU utilization metric id is "/stats/cpu/utilization". For a given metric there can be more than one series of data. For example with the "/stats/cpu/utilization" metric, on a system containing 4 CPU cores this metric will have 4 unique data series, one for each CPU core. The unique parameters that identify each data series are referred to as "contributors". There is always an implicit contributor which is the current router name and usually there is also a node contributor too as metrics are generally generated by a specific node (eg:in the case of the CPU utilization each node will have its own 4 CPU cores). For the CPU metric the contributors are router, node and core and the t128MetricContributors object will contain a string representation of these values so that each row can be correlated with the correct metric data series.

## Configuration File

---

The metrics configuration file is stored in `/etc/128technology/snmpMetricsConfig.json`. There is currently only one element; "metrics" which is an array of metrics configuration elements. Each of these elements represents one metric configuration. The following elements are used to configure each metric:

Element	Required	Description
alias	yes	Base alias returned in <code>t128MetricAlias</code>
metric_id	yes	The modelled metric ID in path form
contributors	yes	The list of metrics contributors defined in the metrics model
filters	no	An array of elements containing contributor/value pairs to filter metric with

## Identifying Metric ID and Contributors

The metrics are modelled in YANG form which is then exported on the 128T device as the file `/var/model/consolidatedStatsModel.xml`. This document contains a hierarchical structure that defines the metrics, for example the `/stats/cpu/utilization` is defined as shown:

```
<yin:container name="stats">
  ...
  <yin:container xmlns:nm="http://128technology.com/t128/stats/node-monitor"
name="cpu" module-prefix="nm" module-name="node-monitor-stats">
  ...
  <yin:container name="utilization">
    <stats:metric-type>meter</stats:metric-type>
    ...
    <yin:leaf name="node">
      ...
    </yin:leaf>
    <yin:leaf name="core">
      ...
    </yin:leaf>
```

Note that some of the XML is omitted for brevity. Within this model it is possible to see the "node" and "core" contributors represented as leafs of the "utilization" metric.

## Example Configurations

The following configurations give some working examples of metrics configurations.

### System and process metrics

The following configuration provides metrics related to the system CPU, disk, memory and process metrics:

```
{
  "metrics": [
    {
      "alias": "cpuUtilization",
      "metric_id": "/stats/cpu/utilization",
      "contributors": ["node", "core"]
    },
    {
      "alias": "memoryCapacity",
      "metric_id": "/stats/memory/capacity",
      "contributors": ["node"]
    },
    {
      "alias": "memoryUsed",
      "metric_id": "/stats/memory/used",
      "contributors": ["node"]
    },
    {
      "alias": "diskCapacity",
      "metric_id": "/stats/disk/capacity",
      "contributors": ["node"]
    },
    {
      "alias": "diskUsed",
      "metric_id": "/stats/disk/used",
      "contributors": ["node"]
    },
    {
      "alias": "processCpuUsage",
      "metric_id": "/stats/process/cpu/usage",
      "contributors": ["node", "process-name"]
    },
    {
      "alias": "processMemoryRss",
      "metric_id": "/stats/process/memory/rss",
      "contributors": ["node", "process-name"]
    }
  ]
}
```

## Aggregate bandwidth and active session count

The following configuration provides metrics showing the aggregate bandwidth by each service, service-route, network-interface and tenant on the 128T platform as well as the total session count:

```
{
  "metrics": [
    {
      "alias": "aggregateBandwidthByService",
      "metric_id": "/stats/aggregate-session/service/bandwidth",
      "contributors": ["service"]
    },
    {
      "alias": "aggregateBandwidthByServiceRoute",
      "metric_id": "/stats/aggregate-session/service-route/bandwidth",
      "contributors": ["service-route"]
    },
    {
      "alias": "aggregateBandwidthByTenant",
      "metric_id": "/stats/aggregate-session/tenant/bandwidth",
      "contributors": ["tenant"]
    },
    {
      "alias": "aggregateBandwidthByNetworkInterface",
      "metric_id": "/stats/aggregate-session/network-interface/bandwidth",
      "contributors": ["network-interface"]
    },
    {
      "alias": "sessionActive",
      "metric_id": "/stats/session/active",
      "contributors": ["node"]
    }
  ]
}
```

## CPU usage filtered by core

Metrics can be filtered to only the series desired. This may be done to minimize the size of the data being polled or to reduce extraneous information. Below is an example of a configuration showing a very simple metrics configuration with only the CPU utilization metric, filtering for metrics from only two of the cores:

```

{
  "metrics": [
    {
      "alias": "cpuUtilization",
      "metric_id": "/stats/cpu/utilization",
      "contributors": ["node", "core"],
      "filters": [{"core": "0"}, {"core": "1"}]
    }
  ]
}

```

The resulting query of the t128MetricsTable is shown below:

```

snmptable -Cl -Ci -mALL -v2c -c public 172.18.0.55 T128-METRICS-
MIB::t128MetricsTable
SNMP table: T128-METRICS-MIB::t128MetricsTable

index                t128MetricContributors    t128MetricValue
"cpuUtilization_0".0  Fabric128.test1.0         1
"cpuUtilization_1".0  Fabric128.test1.1         100

```

Note that in this response the t128MetricAlias returns two metric instances "cpuUtilization\_0" and "cpuUtilization\_1", one for each filter. Within each of these metrics instances there is one data series one for each data series and that the series are limited to only the cores specified (in this system cores 2 and 3 are being filtered).