

## Application Note



## Application Note

# PoE device and port power consumption polling using User Defined polling

6/28/2019

John Diamond - Entuity



## Table of Contents

1	Overview .....	3
2	Setting up the device Total PoE Power consumption attribute and collector.....	4
3	Setting up the port PoE Power consumption attribute and collector .....	6

## 1 Overview

Per-port power consumption figures are available for many PoE-equipped Cisco switches from the CISCO-POWER-ETHERNET-EXT-MIB. There's a table called `cpeExtPsePortTable` that provides a number of metrics including the per-port power consumption in mW. The complication when implementing user defined polling to leverage the OIDs in this table are that it is indexed by a combination of the slot and port number. The port power consumption is available from the `cpeExtPsePortPwrConsumption` OID (1.3.6.1.4.1.9.9.402.1.2.1.9) so the OID for the consumption of slot 2 port 5 would be 1.3.6.1.4.1.9.9.402.1.2.1.9.2.5. The index is effectively 2.5 in this case. This is not a standard available indexing mechanism in ENA which uses the MIB-2 `ifIndex` to identify ports.

One approach to obtaining the `slot.port` index from the `ifIndex` is to first convert the `ifIndex` into an Entity MIB index then convert the Entity MIB index into the `slot.port` index. This can be done using the following tables:

The `entAliasMappingIdentifier` OID (1.3.6.1.2.1.47.1.3.2.1.2.entityMIBIndex.0) allows the `ifIndex` to be looked up given the `entityMIBIndex`. Here's an example of five ports from this table:

```
.1.3.6.1.2.1.47.1.3.2.1.2.1062.0 = OID: .1.3.6.1.2.1.2.2.1.1.8
.1.3.6.1.2.1.47.1.3.2.1.2.1063.0 = OID: .1.3.6.1.2.1.2.2.1.1.9
.1.3.6.1.2.1.47.1.3.2.1.2.1064.0 = OID: .1.3.6.1.2.1.2.2.1.1.10
.1.3.6.1.2.1.47.1.3.2.1.2.1065.0 = OID: .1.3.6.1.2.1.2.2.1.1.11
.1.3.6.1.2.1.47.1.3.2.1.2.1066.0 = OID: .1.3.6.1.2.1.2.2.1.1.12
```

This shows that `entityMIBIndex` 1062 corresponds to `ifIndex` 8 and that 1063 corresponds to 9, etc.

The `cpeExtPsePortEntPhyIndex` OID (1.3.6.1.4.1.9.9.402.1.2.1.11.slot.port) allows the `entityMIBIndex` to be looked up given the `slot.port` index. Here's an example of five ports from this table:

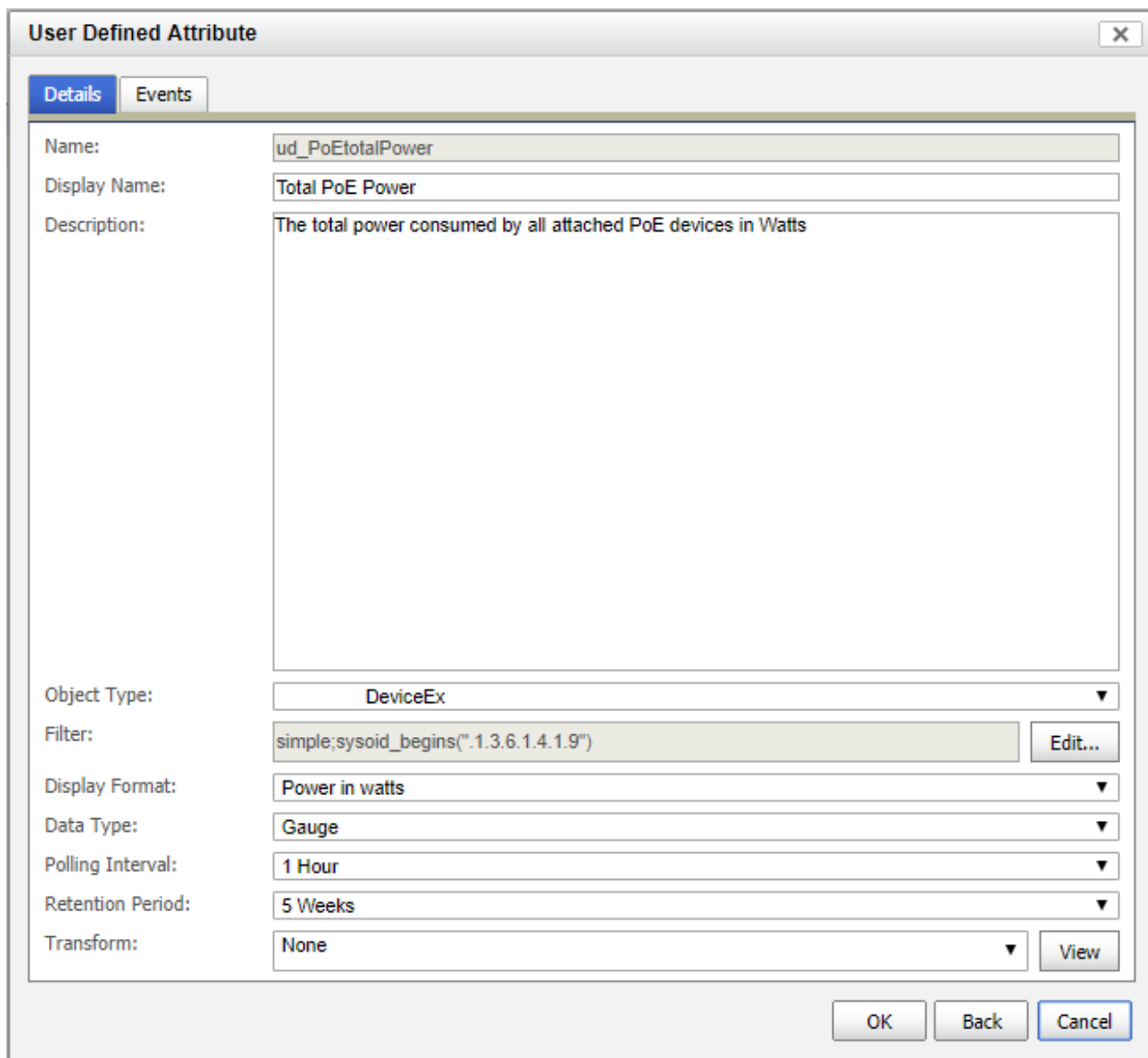
```
.1.3.6.1.4.1.9.9.402.1.2.1.11.1.1 = INTEGER: 1062
.1.3.6.1.4.1.9.9.402.1.2.1.11.1.2 = INTEGER: 1063
.1.3.6.1.4.1.9.9.402.1.2.1.11.1.3 = INTEGER: 1064
.1.3.6.1.4.1.9.9.402.1.2.1.11.1.4 = INTEGER: 1065
.1.3.6.1.4.1.9.9.402.1.2.1.11.1.5 = INTEGER: 1066
```

This shows that `slot.port` 1.1 corresponds to `entityMIBIndex` 1062, 1.2 corresponds to 1063, etc.

The primary challenge is that if these conversions were to be performed at the port level it would require two table walks per port which would be an unacceptable runtime overhead. The approach shown below implements a device level attribute that computes the aggregate consumption of all PoE enabled ports. However, before performing this aggregation based on a walk of the table that provides the per port

consumption statistics it creates stash entries for both ifIndex to entityMIBIndex and ifIndex to slot.port index. The first of these stashes is used to make the creation of the second stash more efficient. The second stash is then used at runtime by the port level port consumption collector to look up the slot.port index based on the ifIndex which is the only index available to it.

## 2 Setting up the device Total PoE Power consumption attribute and collector



**User Defined Attribute**

**Details** | Events

Name:

Display Name:

Description:

Object Type:

Filter:

Display Format:

Data Type:

Polling Interval:

Retention Period:

Transform:

**Name:** ud\_PoEtotalPower

**Display Name:** Total PoE Power

**Description:** The total power consumed by all attached PoE devices in Watts

**User Defined Collector (User Defined Attribute)**

Name:

Description:

Object Type:

Attribute:

OID:

Index:

SNMP Version:  v1  v2 (Note: This setting is ignored if the device is being managed as v3 or v2c only)

Method:

Filter:

Priority:

Transform:

**Name:** ud\_PoEtotalPower

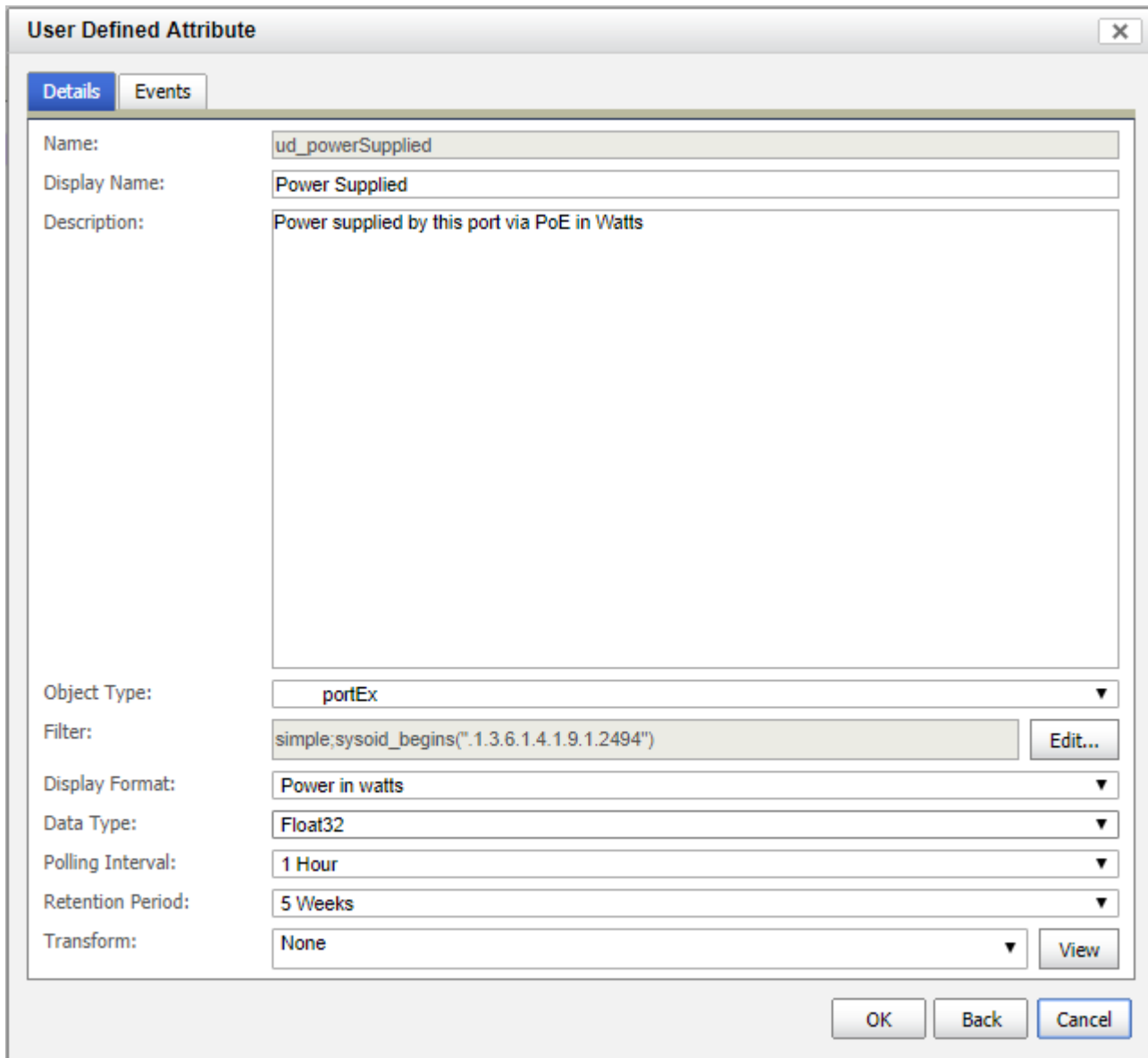
**Description:** The total power consumed by all attached PoE devices in Watts

**Method:** simple;variable

```
deviceId=id;foreach(snmp_walk(snmpv2, ".1.3.6.1.4.1.9.9.402.1.2.1.11", 1000), stash_put(concat("Entity_to_PoE_Index:", deviceId, ":", this[1]), this[0]));foreach(snmp_walk(snmpv2, ".1.3.6.1.2.1.47.1.3.2.1.2", 1000), stash_put(concat("PoE_Index:", deviceId, ":", substringoid(this[1], 10, 11)), stash_get(concat("Entity_to_PoE_Index:", deviceId, ":", substringoid(this[0], 0, 1))));stash_exists(concat("Entity_to_PoE_Index:", deviceId, ":", substringoid(this[0], 0, 1)));convert(sum(foreach(snmp_walk(snmpv2, ".1.3.6.1.4.1.9.9.402.1.2.1.9", 2000), this[1])) / 1000, uint32)
```

**Filter:** simple;sysoid\_begins(".1.3.6.1.4.1.9")

### 3 Setting up the port PoE Power consumption attribute and collector



**User Defined Attribute**

**Details** | Events

Name: ud\_powerSupplied

Display Name: Power Supplied

Description: Power supplied by this port via PoE in Watts

Object Type: portEx

Filter: simple;sysoid\_begins(".1.3.6.1.4.1.9.1.2494") Edit...

Display Format: Power in watts

Data Type: Float32

Polling Interval: 1 Hour

Retention Period: 5 Weeks

Transform: None View

OK Back Cancel

**Name:** ud\_powerSupplied

**Display Name:** Power Supplied

**Description:** Power supplied by this port via PoE in Watts

**User Defined Collector (User Defined Attribute)** X

Name:

Description:

Object Type:

Attribute:

OID:

Index:

SNMP Version:  v1  v2 (Note: This setting is ignored if the device is being managed as v3 or v2c only)

Method:

Filter:

Priority:

Transform:

**Name:** ud\_powerSupplied

**Description:** Power supplied by this port via PoE in Watts

**Method:**

simple;variable

PoE\_Index=stash\_get(concat("PoE\_Index:",ref.device.id,":",portIndex.ifIndex));snmp\_get(snmv2, ".1.3.6.1.4.1.9.9.402.1.2.1.9", PoE\_Index) / 1000.0

**Filter:** simple;stash\_exists(concat("PoE\_Index:",ref.device.id,":",portIndex.ifIndex))