opCharts Polling Definition

Polling definitions describe what will be polled when a collector is running, and how it will be presented. In the first release of real-time charts, polling definitions are imported/exported via opcharts-cli and can be viewed in the GUI.

opCharts ships with several polling definitions that may be useful and can be used as a guide when creating new ones. These files can be found in omk /install/nmisd/pollers.d/

Polling definitions must define their name (which must be unique) and can optionally have a description. There are 5 main sections in a polling definition (see below).

Changing a polling definition is not currently supported, if you need to do this please create a new poller with a new name and start collectors using the new poller.

Adding/Importing, Exporting, Removing

To add/import new polling definitions use opcharts-cli.pl, specify the file to get the definition from and if the file has more than one, specify the name of the poller to import:

```
/usr/local/omk/bin/opcharts-cli.pl act=import-pollers file=/usr/local/omk/install/nmisd/pollers.d/my_new_poller.
json
```

Exporting takes the current poller definition and writes it to the file specified (or to the screen if no file is given):

```
/usr/local/omk/bin/opcharts-cli.pl act=export-pollers name="My Export Poller" file=/tmp/poller_1.json
```

Removing a poller requires that no collectors exist that use it (enabled or not). After all the collectors that use the poller have been removed, delete the poller like this:

```
/usr/local/omk/bin/opcharts-cli.pl act=delete-pollers name="My Old Poller"
```

Sections

1. Graph options - set graph attributes (like axis titles, min/max).
2. Properties - values that generally don't change over time but are needed to calculate a field or just presented with the graph.
3. Virtual Properties - definitions of new variables which can be calculations using properties, other virtual properties and collector parameters.
4. Fields - values that will be polled every time the collector runs.
5. Virtual Fields - definitions of new variables which can be calculations using fields, properties, virtual properties and collector parameters.

Graph Options

The following are options that can be set:

```
'titleText'
'subtitleText'
'yAxis0TitleText'
'yAxis1TitleText'
'yAxis0Min'
'yAxis0Max'
'yAxis1Min'
'yAxis1Max'
```

Common attributes (Properties & Fields)

Properties and Fields define data that will be polled, because of this their definitions share common attributes. The following attributes are common to both:

```
"name" : "name of property/field, if not defined the snmp_mib name will be used",
"data_type" : "gauge/counter",
"snmp_oid" : "the oid to be polled, if not indexed .0 will be appended",
"snmp_mib" : "name of mib",
"show_in_chart": true/false,
"indexed": true/false, if true the index given in the collection definition will be appended to the snmp_oid,
otherwise .0 will be appended
```

Common attributes (Virtual Properties & Virtual Fields)

Virtual sections all run through the same parsing system so their definitions are similar, see the opCharts Resource Model Description Virtual Options section for more information.

The differences between property and field definitions is largely the data that is available to them when running calculations.

Properties

In addition to the common property/field attributes, properties can/must have the following attributes:

```
"property_type" : "snmp", must be defined, snmp is currently the only option
```

Virtual Properties

See the common virtual property/field attributes.

Fields

In addition to the common property/field attributes, fields can/must have the following attributes:

```
"field_type" : "snmp", must be defined, snmp is currently the only options

"axis": 0/1, which axis to put the data on, defaults to 0 if not defined
"colour": "#aabbcc", colour to assign the dataset, will be automatically picked if not defined
"decimals": 2, the number of decimal places to show
"display_order": 1, defines the order the data should be displayed in the chart
"reverse_axis": false/true, flips the values (multiplies by -1) so they show below the X axis
"stack": 1, if not present, stacking off, if present defines the stack the data will appear in, think of it
like a group
"suffix": " string", will be appended as a suffix to the value when looking at a point on the chart
"type": "line/area", defaults to line if not present
```

Virtual Fields

In addition to the common virtual property/field attributes, it can also have all of the attributes defined in Fields (field_type can be defined but will not do anything)

Example

Here is an example to help make the definition clearer:

```
[
        {
                "name" : "Cisco Memory Usage",
                "description" : "Cisco Memory Polling, for processor memory",
                "properties": [
                        {
                                "property_type" : "snmp",
                                "data_type" : "string",
                                "name" : "ciscoMemoryPoolName",
                                "snmp_oid" : "1.3.6.1.4.1.9.9.48.1.1.1.2.1",
                                "snmp_mib" : "ciscoMemoryPoolName",
                                "show_in_chart": false,
                                "indexed": false
                        }
                ],
                "virtual_properties": [
                        {
                                "name" : "useless virtual proplerty",
                                "operation" : "calculation",
                                "value" : "$ciscoMemoryPoolName + $collector_frequency"
                        },
                ],
                "fields": [
                        {
                                "field_type" : "snmp",
                                "data_type" : "gauge",
                                "name" : "ciscoMemoryPoolUsed",
                                "snmp_oid" : "1.3.6.1.4.1.9.9.48.1.1.1.5.1",
                                "snmp_mib" : "ciscoMemoryPoolUsed",
                                "indexed": false,
                                "indexed_comment" : "if not indexed, the system should append .0 to the end of
the OID",
                                "show_in_chart": false
                        },
                        {
                                "field_type" : "snmp",
                                "data_type" : "gauge",
                                "name" : "ciscoMemoryPoolFree",
                                "snmp_oid" : "1.3.6.1.4.1.9.9.48.1.1.1.6.1",
                                "snmp_mib" : "ciscoMemoryPoolFree",
                                "indexed": false,
                                "indexed_comment" : "if not indexed, the system should append .0 to the end of
the OID",
                                "show_in_chart" : false
                        }
                ],
                "virtual_fields" : [
                        {
                                "name" : "ciscoMemoryPoolTotal",
                                "operation" : "calculation",
                                "value" : "$ciscoMemoryPoolUsed + $ciscoMemoryPoolFree"
                        },
                        {
                                "name" : "MemoryUtilisation",
                                "operation" : "calculation",
                                "value" : "$ciscoMemoryPoolUsed / $ciscoMemoryPoolTotal * 100",
                                "suffix" : "%",
                                "show_in_chart": true,
                        }
                ],
                "graph_options" : {
                  "title" : "$node Memory Utilisation",
                  "yAxis0TitleText" : "Memory Usage %",
                  "yAxis0Max" : 100,
                  "yAxis0Min" : 0
                }
        }
]
```