



EMC® Watch4net®

Version 6.3u2

Index Generator Filter Admin Guide

P/N: pending

Rev : 40030

EMC²

Copyright ©2009 - 2014 EMC Corporation. All rights reserved. Published in the USA.

Published April, 2014

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided as is. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC², EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries. All other trademarks used herein are the property of their respective owners.

For the most up-to-date regulatory document for your product line, go to EMC Online Support (<https://support.emc.com>).

Contents

Preface	iii
Purpose	iii
Intended Audience	iii
Prerequisites	iii
Document Organization	iv
Common Abbreviations and Acronyms	iv
Where to get help	v
1 Description	1
2 Installation	3
3 Configuration	5
1 Configuration example	5
2 Usage example: variable index hashcode suffix generation	8
3 Usage example: full variable index generation	11
4 Known limitations	13

Preface

Purpose

The purpose of this guide is to provide configuration and usage information about the Index Generator Filter.

Intended Audience

APG administrators and system integrators, as well as users responsible of managing data collection configuration should read this document.

Prerequisites

- The *APG* collector manager must be installed. This filter module may be used with *APG* versions 5.0 and later.

Document Organization

1. Description	A description of this filter function.
2. Installation	Instructions on how to install this filter.
3. Configuration	Instructions on how to configure this filter.
4. Known limitations	A list of known limitations.

Common Abbreviations and Acronyms

Not Applicable

There are no abbreviations used in this document.

Where to get help

EMC support, product, and licensing information can be obtained as follows:

PRODUCT INFORMATION - For documentation, release notes, software updates, or information about EMC products, go to EMC Online Support at:

<https://support.emc.com>

TECHNICAL SUPPORT - Go to EMC Online Support and click Service Center. You will see several options for contacting EMC Technical Support. Note that to open a service request, you must have a valid support agreement. Contact your EMC sales representative for details about obtaining a valid support agreement or with questions about your account.

Description

The IndexGenerator filter is used to generate a new index for a raw value, using its properties. Two modes of index generation are supported:

- It can truncate the variable part of the raw value, from the last occurrence of a specified delimiter character, then appends a *hashcode suffix* based on the property values set in the raw value to the variable. The hashcode part of the variable is the new index.
- It can generate a full *new variable index* when no delimiter character is specified, based on the property values set. In this case, the values of the properties used to generate the index will appear in clear text.

The properties to use in order to generate the variable *hashcode suffix* or the full *variable index* are specified in the filter configuration. The raw values affected by this variable transformation are specified by matching the value of their “name” property.

This filter can be used with any collector when generating a correct index (variable) is not possible due to model complexity, incoherences or collecting specification limitations.

It can also be used to generate a variable suffix when pushing raw values from a script.

2

Installation

The Index Generator Filter module may be installed like any other module. The package name is: "index-generator-filter". Use the module manager to install this module.

Then, edit the APG/Collecting/Collector-Manager/Default/conf/collecting.xml file (location may vary depending upon installation location and Collector-Manager instance name) and insert the following line in the existing configuration. Please refer to the *APG* Backend Administration Guide if unsure on how to do this.

```
<?xml version="1.0" encoding="UTF-8"?>
<config ...>
  <connectors>
    ... connector already defined ...
  </connectors>
  <filters>
    ... next filter ...
    <filter enabled="true" name="Index-Generator-Filter" next="NextFilter"
      config="Index-Generator-Filter/Default/conf/index-generator.xml" />
    ... previous filter ...
  </filters>
```

The “index-generator.xml” configuration file should be located in the APG/Collecting/Index-Generator-Filter/Default/conf/ directory. This file will contain your customized configuration for this filter. Edit it as explained in this document’s Configuration section. Then restart the *APG* Collector Manager service.

Configuration

This section contains configuration examples as well as the description of the available configuration elements.

1 Configuration example

Let's use the following example in order to create a configuration for this filter:

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:config xmlns:tns="http://www.watch4net.com/index-generator" xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance" xsi:schemaLocation="http://www.watch4net.com/index-generator_index-
generator.xsd">
  <tns:rawvalue-names>
    <tns:name>metricName</tns:name>
  </tns:rawvalue-names>
  <tns:index-generator-properties>
    <tns:property>property</tns:property>
  </tns:index-generator-properties>
  <tns:index-is-after-last-char>.</tns:index-is-after-last-char>
</tns:config>
```

Elements and attributes are as follows:

- `<tns:rawvalue-names>` Contains a list of “name” property matching values. The “name” property is used to identify the raw values modified by the filter.
- `<tns:name>metricName</tns:name>` The raw value “name” property. At least one must be specified.
- `<tns:index-generator-properties>` Contains a list of property names. The values linked to these names are used to generate the new variable index.
- `<tns:property>property</tns:property>` The index generating property to use in the raw value. At least one must be supplied.
- `<tns:index-is-after-last-char>.</tns:index-is-after-last-char>` The character to use as the delimiter. Anything beyond this character last occurrence will be stripped from the raw value variable and replaced by the generated variable *hashcode suffix*. This element is optional: a full *variable index* will be generated if it is not provided.

Note: The `<tns:index-is-after-last-char>.</tns:index-is-after-last-char>` configuration element is especially useful when dealing with raw values generated from the SNMP Collector. The SNMP Collector raw value variable index is made up of several fields, separated by period characters. The last field, after the last period character, is a hashcode generated by the SNMP Collector. It is sometimes necessary to replace this hashcode with another *hashcode suffix*, generated using several property values, to better identify the generated raw values.

Note: In some cases, the generated raw values are linked to their collector by their variable index. The Index Generator Filter may be used to break this link by generating a full index which does not contain the collector name. This is useful in highly distributed collecting architectures, where the collector-to-device relationship may change over time.

2 Usage example: variable index hashcode suffix generation

Let suppose that the raw values going through this filter have the following properties:

- device
- part
- subpart
- name

And that the variable index for these raw values is a static value: `valueFromMyCollector.0`

Note that it is not a good idea to send raw values like these directly to the backend, as they all have the same variable. This filter provides a work around, by generating a suffix which will be concatenated to the original variable index.

If the raw value has these properties set up:

- device=DiskArray17
- part=Cabinet3
- subpart=Disk3
- name=SpaceLeft

The raw value will look like this before it is processed by the filter:

```
+r 1199145600 group valueFromMyCollector.0 100 device=DiskArray17
name=SpaceLeft subpart=Disk3 part=Cabinet3
```

The variable index part of the raw value is the “`valueFromMyCollector.0`” string.

The filter generates a *hashcode suffix* similar to:

```
hashcode(DiskArray17.Cabinet3.Disk3.SpaceLeft)
```

The resulting raw value, after processing, will look like this:

```
+r 1199145600 group valueFromMyCollector.-875434 100 device=DiskArray17
name=SpaceLeft subpart=Disk3 part=Cabinet3
```

Note: The `.0` suffix was replaced by a hashcode, generated from the concatenated property values “DiskAr-

ray17.Cabinet3.Disk3.SpaceLeft" The hashcode is signed 32-bit integer.

This implies that collecting scripts can be simplified and simply push raw values with a generic variable index: the filter will take care of renaming them correctly as long as the properties are provided in the raw value. The configuration required change the raw value variable index is displayed on the next page.

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:config xmlns:tns="http://www.watch4net.com/index-generator" xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance" xsi:schemaLocation="http://www.watch4net.com/index-generator_index-
generator.xsd">
  <tns:rawvalue-names>
    <tns:name>SpaceLeft</tns:name>
  </tns:rawvalue-names>
  <tns:index-generator-properties>
    <tns:property>device</tns:property>
    <tns:property>part</tns:property>
    <tns:property>subpart</tns:property>
    <tns:property>name</tns:property>
  </tns:index-generator-properties>
  <tns:index-is-after-last-char>.</tns:index-is-after-last-char>
</tns:config>
```

3 Usage example: full variable index generation

Again, let's suppose that the raw values going through this filter have the following properties:

- device=WirelessController-Alpha
- part=AccessPoint-1
- name=SignalToNoiseRatio
- mac=0x00 0x05 0x04 0x03 0x02 0x01
- ip=10.128.0.1
- source=W4NSnmp-1

In this case however, we do not care about the original variable name and do not want to use it in the backend. Specifically, we do not want the “ip” property value to appear anywhere in the variable (the ip property value is always a part of the variables generated by the SNMP collector as it is used to identify the device on which the raw value was collected).

The original index (variable) would look like this:

```
W4NSnmp-1.10.128.1.1.radioSignal..1.3..1723
```

This indicates that variable's source is W4NSnmp-1, the ip of device on which the raw value was collected is 10.129.1.1, the table id where the raw value comes from is radioSignal, its column is .1.3, the metric was collected as a COUNTER (nothing specified) and its unit is not set.

Using this configuration will generate a unique variable index for this metric. But instead of a variable linked to the device, it would be linked to the mac address only (no source is required in this case as the MAC is a unique global identifier):

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:config xmlns:tns="http://www.watch4net.com/index-generator" xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance" xsi:schemaLocation="http://www.watch4net.com/index-generator_index-
generator.xsd">
  <tns:rawvalue-names>
    <tns:name>SignalToNoiseRatio</tns:name>
  </tns:rawvalue-names>
  <tns:index-generator-properties>
    <tns:property>mac</tns:property>
    <tns:property>name</tns:property>
  </tns:index-generator-properties>
</tns:config>
```

The following variable will result from this transformation:

0x00 0x05 0x04 0x03 0x02 0x01.SignalToNoiseRatio

4

Known limitations

Any variable longer than 255 characters will be stripped of extra information. In this case, the backend may confuse two variables which start with the first 255 characters. Please use a shorter index in this case.