



EMC® Watch4net®

Version 6.3u2

Value Offset Filter Admin Guide

P/N: pending

Rev : 40030

EMC²

Copyright ©2010 - 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
Where to get help	v
1 Description	1
2 Installation	3
3 Configuration	5
1 Configuration example	5
2 Configuration elements description	6
3 Configuration example explanation	7
4 Troubleshooting	9

Preface

Purpose

The purpose of this guide is to provide configuration and usage information about the *Value Offset 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. Troubleshooting	A list of common questions and answers about this filter.

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.

1

Description

This filter provides an easy way to recalculate values produced by any APG collector. It may be used to increase or decrease the ICMP echo round trip time by a constant amount, change the value scale by applying a scale factor (through multiplication or division) or raise the value to the power of a specified argument. The values selected for processing are calculated within the collecting process, therefore eliminating some calculations at report generation. This filter can also eliminate values if they are below or above a specified value.

Note: Any non-numerical value received as input by this filter, or resulting from it's application, will not be pushed to the next registered filter/connector component. This includes NaN (Not a number) and Infinity. Also, specifying unsupported operations will prevent values from reaching the next registered filter/connector component. Therefore, this filter should be located after any other filter which uses NaN values to indicate a failure condition, such as the PingFilter which comes with the IcmpCollector.

2

Installation

The *Value Offset Filter* module may be installed like any other module. The package name is: “value-offset-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>
    <filter enabled="true" name="NextFilter" next="..."
      config="..." />
    <filter enabled="true" name="MyFilter" next="NextFilter"
      config="Value-Offset-Filter/Default/conf/MyValueOffsetFilterConf.xml" />
    <filter enabled="true" name="PreviousFilter" next="MyFilter"
      config="..." />
  </filters>
```

The “MyValueOffsetFilterConf.xml” configuration file should be located in the APG/Collecting/Value-Offset-Filter/Default/conf/ directory. This file will contain your customized configuration for this filter. Create and edit as explained in this document’s Configuration section. Then restart the *APG* Collector Manager service.

3

Configuration

This section contains a configuration example 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:value-offset-filter-config xmlns:xs="http://www.watch4net.com/value-offset-filter"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.watch4net
.com/value-offset-filter_value-offset-filter.xsd">
  <xs:offsets-by-source source="testSource">
    <xs:offset name="testVariable">
      <xs:simple-operation>
        <xs:operator>+</xs:operator>
        <xs:value>73.0</xs:value>
      </xs:simple-operation>
    </xs:offset>
  </xs:offsets-by-source>
</tns:value-offset-filter-config>
```

```
</xs:offsets-by-source>
<xs:offsets-by-source source="testSource2">
  <xs:offset name="testVariable2">
    <xs:simple-operation>
      <xs:operator>-</xs:operator>
      <xs:value>40.0</xs:value>
    </xs:simple-operation>
  </xs:offset>
  <xs:offset name="testVariable8">
    <xs:simple-operation>
      <xs:operator>X</xs:operator>
      <xs:property>multiplier</xs:property>
    </xs:simple-operation>
  </xs:offset>
</xs:offsets-by-source>
<xs:offset-sourceless>
  <xs:stand-alone-offset name="testVariable3">
    <xs:simple-operation>
      <xs:operator>]</xs:operator>
      <xs:value>1.0</xs:value>
    </xs:simple-operation>
  </xs:stand-alone-offset>
  <xs:stand-alone-offset name="testVariable4">
    <xs:simple-operation>
      <xs:operator>r</xs:operator>
      <xs:property>usevalue</xs:property>
    </xs:simple-operation>
  </xs:stand-alone-offset>
</xs:offset-sourceless>
</xs:value-offset-filter-config>
```

Note: Please use an XML editor which supports XML schema validation to edit this configuration file.

2 Configuration elements description

Either a `<xs:offsets-by-source>` or `<xs:offset-sourceless>` elements must be defined to use this filter. Both types of elements may be present in the same configuration.

When using `<xs:offsets-by-source>`, the source attribute is used to specify the collecting source

of the raw values that will be accepted by the filter. In this example, the collector which points to this filter (in the collecting.xml file, see below) must have its source attribute set to “testSource”, otherwise the values it generates will simply bypass the filter. Several sources may be specified when more than one collector points to this filter.

The element `<xs:offset-sourceless>` must contain at least one `<xs:stand-alone-offset>` element, which is defined like the `<xs:offset>` element described below.

`<xs:offsets-by-source>` : Must contain at least one `<xs:offset>` element. The name attribute is used to specify the target variable name property. In this case, all raw values that have a name property set to “testVariable” will be processed by this filter.

`<xs:offsets>` : Must contain one `<xs:simple-operation>` element. This element describes the calculation to apply on the raw value’s value. It must contain the following elements:

`<xs:simple-operation>` : specifies the operator to apply on the value. Allowable operators are:

- + : Addition
- - : Subtraction
- * : Multiplication
- / : Division
- ^ : raise to the power of X
- r : replace the value by a property value or a hardcoded value.

Other operators include the following inequalities:

- l (less than)
- m (more than)
- [(less or equal than)
-] (more or equal than)

These inequalities will filter out variables with values that do not meet the criteria specified in the `<xs:value>` element.

`<xs:value>` : specifies the X constant (floating point type).

`<xs:property>` : specifies the property which contains the X constant.

3 Configuration example explanation

- All raw values that have their source property set to “testSource” and name property set to “testVariable” will have their value increased by 73.0.

- All raw values that have their source property set to "testSource2" and name property set to "testVariable2" will have their value decreased by 40.0.
- All raw values that have their source property set to "testSource2" and name property set to "testVariable8" will have their value multiplied by the value found in their "multiplier" property.
- All variables with their name property set to testVariable3 and have a value more or equal to 1.0 will be let through by the filter. Values less than 1 will be rejected by the filter.
- All variables with their name property set to testVariable4 will have their value replaced by the value contained in their "usevalue" property. The value contained in the property must be numeric of course, otherwise the value is not replaced and the raw value is sent as is.

4

Troubleshooting

No data is collected

- Make sure the ValueOffsetFilter filter is enabled and that it points to the correct next component in the collecting.xml file. Also ensure that the other plugins and filter s are enabled in this data path.
- Make sure the specified `<xs:operation>` is supported. Supported operations are described in the [2](#) section.
- Make sure a division by zero is not performed: this results in an infinite value, which will not be pushed to the next registered collecting component.

The collected raw values are not modified

- Make sure the ValueOffsetFilter's intended `<xs:offsets-by-source>` source attribute matches the collector's source attribute.

- Check the `<xs:offset>` name attribute and make sure it matches the target raw values' name property.

The collected raw values are not modified properly

- Make sure the specified `<xs:value>` contains a value. A value of 0 is assumed when no value is provided. This value can have several effect, such as not modifying the value (for the +, - operators), always giving out zero as a result (for the multiplication operator), or 1 when raising the value to the power of 0...
- Rounding errors may be present. This is normal since values use floating point arithmetics.
- If the named variable appears in one of the `<xs:offsets-by-source>` element and a `<xs:offset-sourceless>` element, both calculations will be applied in that order. (source, then sourceless).