



EMC[®] Watch4net[®]

Version 6.3u2

Cross Referencing Filter Admin Guide

P/N: pending

Rev : 45574

EMC²

Copyright ©2008 - 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
Intended Audience	iii
Prerequisites	iii
Where to get help	v
1 Product Overview	1
2 Architectural Overview	3
3 Things to do before installation	5
4 Installation	7
1 APG 4.1 and under	7
2 APG 4.2 and over	7
5 Configuration	9
1 Collecting service configuration	9
2 Filter configuration	10
6 Things to do after configuration	13
7 Appendix A	15
1 Configuration Example	15

Preface

The purpose of this guide is to provide information about the Cross Referencing Filter for APG. It contains all the necessary information to install and configure the filter.

Intended Audience

This guide is intended to be read by any *APG* administrator who will configure the Cross Referencing Filter to tag values from different sources with the same set of properties.

Prerequisites

It is assumed that the *APG* application has been properly installed and configured on the server.

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.

Product Overview

Administrators that use APG to collect data from different sources sometimes come across situations where values that belong to the same device or part are not tagged with the same properties. For example, some metrics about a host might be collected using one collector while other metrics on the same host are collected by a different collector. If the two different data sources do not contain the same information, it is possible that the metrics from the first collector are tagged with more properties than metrics from the second one. This can be a problem when comes the time to design unified reports, because some metrics might not appear depending on the way reports are built.

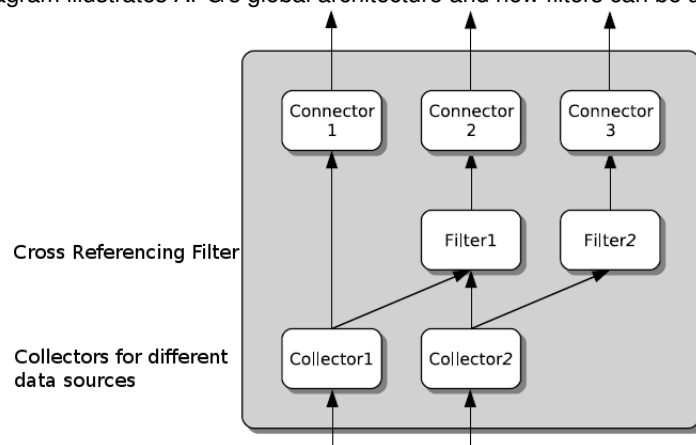
The Cross Referencing Filter for APG provides a solution to this problem by allowing metrics to be tagged with properties that belong to another metric, based on fully customizable match criteria. The Cross Referencing Filter has no dependencies to any other APG libraries. However, since Java 6 is needed, the following requirements must be met:

- APG 4.0 or higher

2

Architectural Overview

The Cross Referencing Filter is built to integrate seamlessly into the APG structure. The following diagram illustrates APG's global architecture and how filters can be attached to it.



3

Things to do before installation

The following operations *must* be performed before any installation or configuration steps are taken.

1. Backup the following file:

- <APG Root>/APG-Backend/conf/collecting.xml

2. Shut down the APG collecting service

4

Installation

1 APG 4.1 and under

To install the Cross Referencing filter, simply copy the following files to the specified location:

File	Destination
backend-collector-dcf-crossreference	<APG-Root>/APG-Backend/lib/
cross-reference-filter.xsd	<APG-Root>/APG-Backend/conf/

2 APG 4.2 and over

An up-to-date version of the Cross Referencing Filter is already provided with your APG distribution. Skip the installation step and go directly to configuration (section [5](#))

5

Configuration

This section describes the different configuration files for the filter. All these files are located in the `<APG Root>/APG-Backend/conf` directory. In case some of these files do not exist, they must be created in this folder as well.

1 Collecting service configuration

The Cross Referencing Filter for APG must be added in the main collecting configuration file (`collecting.xml`). Add the following section if it is not present, and make sure that the *enabled* attribute is set to *true*:

```
<filters>
  ...
  <filter enabled="true" name="CrossReferencingFilter" next="Alerting_Backend" config="Cross-
Referencing-Filter/Default/conf/cross-referencing-filter.xml"/>
  ...
</filters>
```

2 Filter configuration

This file is the main configuration file for the filter. It contains information about the way metrics will be matched and the properties that will be added when a match is found. It is up to the administrator to decide how to name this file and to make sure that the selected name is used to configure the filter in `collecting.xml` (see section 1).

Global structure Here the global structure of the configuration file:

- `<configuration>`: The main container for the collector configuration. Every other tag must be nested inside this one.
 - `<blind-write>`: Whether or not the filter will write properties in the data structure every time a new set of key properties is found. If every metric from a same source contains the same value properties for a given set of key properties (properties on which the matching is performed), it is probably a good idea to set this to *false*. Otherwise, it should be left to *true*.
 - `<data-accessor>`: The type of media inside which the properties will be kept. The only existing implementation so far is the *MemoryHashDataAccessor*, which simply keeps data in RAM.
 - `<reference>`: This tag acts as a container for all configurations that relate to the same cross referencing pattern.
 - * `<condition [type=string|filter][property=property]>`: Sets a condition to the matching pattern. The condition can be expressed either as a plain string or as an APG filter expression. The *property* attribute represents the name of the property to which the condition applies when using the string mode. This tag can be repeated as many times as necessary. When multiple conditions are set, a value will need to match all of them in order to be processed.
 - * `<match-on>`: Adds a property on which two metrics must match. This tag can be repeated as many times as necessary.
 - * `<propagate-on>`: When set, this is the list of properties, on a given metric, that will be used to propagate the properties to tag on another match. For example, when `<match-on>` is set to *device* and `<propagate-on>` is set to *reference*, a value containing both *device* and *reference* will update both matches with its own properties.
 - * `<pull-from>`: When set, this is the list of properties, on a given metric, that will be used to retrieve the properties to tag from another match. For example, when `<match-on>`

is set to *device* and *<pull-from>* is set to *referenced*, a value containing both *device* and *referenced* will be updated with the properties from both matches.

- * *<include-properties>*: If this parameter is set to *true*, the properties listed between the *<get-property>* tags will be kept. If it is set to *false*, all properties except those listed in the *<get-property>* tags will be taken.
- * *<expiration>*: expiration of cache in seconds for each value in each entry. Setting a null value or zero and below disable expiration.
- * *<get-property>* A property that must be kept or excluded, depending on the *include-properties* parameter. If and only if the property must be kept (i.e. *include-property* is set to *true*) this property may have an optional attribute in the form of *default=defaultValue*. The *defaultValue* will be used to tag the specified property in the case when the filter does not have a value for this property. An attempt to set the *default* attribute for the excluded property (i.e. *include-property* is set to *false*) will result in an invalid configuration.

6

Things to do after configuration

When the collector is installed and configured, APG is ready to collect new data from the XML source. The following steps can be taken:

1. Start the APG collecting service.
2. Look at the collecting logs to make sure that everything is starting up as it should.

Appendix A

1 Configuration Example

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xmlns="http://www.watch4net.com/CrossReferencingFilter"
               xsi:schemaLocation="http://www.watch4net.com/CrossReferencingFilter/cross-referencing-
filter.xsd">

  <blind-write>true</blind-write>
  <data-accessor>
    com.watch4net.apg.v2.collector.plugins.crossreferencingfilter
    .data.mem.MemoryHashDataAccessor
  </data-accessor>
  <reference>
    <condition property="devtype">Router</condition>
    <match-on>device</match-on>
    <include-properties>true</include-properties>
```

```
<get-property>model</get-property>
<get-property>devdesc</get-property>
</reference>

<reference>
  <condition type="filter">devtype=='Router'&parttype=='Interface'</condition>
  <match-on>device</match-on>
  <match-on>part</match-on>
  <include-properties>true</include-properties>
  <get-property>partdesc</get-property>
  <get-property>ifacemac</get-property>
</reference>
</configuration>
```

Based on the configuration shown above, let's assume that the following metrics are collected:

Metric	Property Name	Property Value
ifInOctets	devtype	Router
	device	W4N-RTR-01
	parttype	Interface
	part	Port 1/1/1
	devdesc	Router in the server room
	model	Generic Broadband Router
	partdesc	Link to Peter's office
ifOutOctets	ifacemac	00:00:00:00:00:42
	devtype	Router
	device	W4N-RTR-01
	parttype	Interface
	part	Port 1/1/1
Availability	devtype	Router
	device	W4N-RTR-01

The resulting metrics will be:

Metric	Property Name	Property Value
ifInOctets	devtype	Router
	device	W4N-RTR-01
	parttype	Interface
	part	Port 1/1/1
	devdesc	Router in the server room
	model	Generic Broadband Router
	partdesc	Link to Peter's office
ifOutOctets	ifacemac	00:00:00:00:00:42
	devtype	Router
	device	W4N-RTR-01
	parttype	Interface
	part	Port 1/1/1
	devdesc	Router in the server room
	model	Generic Broadband Router
Availability	partdesc	Link to Peter's office
	ifacemac	00:00:00:00:00:42
	devtype	Router
	device	W4N-RTR-01
	devdesc	Router in the server room
	model	Generic Broadband Router