



EMC® Watch4net®

Version 6.3u2

Trap Receiver Administration Guide

P/N: pending

Rev : 40032

EMC²

Copyright ©2012 - 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 Config example	5
2 Rules configuration	7
3 Conversion examples	8
4 A complete configuration example	8
5 Reconfiguration	12

Preface

Purpose

The purpose of this guide is to provide configuration and usage information about the Trap Receiver.

Intended Audience

APG administrators and system integrators should read this document.

Prerequisites

- The *APG* event processing manager must be installed. This processing module may be used with *APG* versions 6.1 and later.
- If traps will be received on a port below 1024, this module must be run as root (Administrator).

Document Organization

1. Description	A description of the functions of this processing element.
2. Installation	Instructions on how to install this processing element.
3. Configuration	Instructions on how to configure this processing element.

Common Abbreviations and Acronyms

<APG-BASE>	refers to the APG installation directory
<INSTANCE>	refers to the instance name of the module (typically "Default")

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

The Trap Receiver is a processing element designed to receive and forward SNMP Traps. It is capable of receiving traps using any of the SNMP protocols (up to and including v3). It is capable of forwarding traps in the same format it received them, or transforming them into Events to be handled by the Event Processing Manager. Thus this component may be used either to receive and process traps, or simply to forward them to their eventual destination.

2

Installation

The Trap Receiver module may be installed like any other module. The package name is: “trap-receiver”. You can install it using the module-manager with the following command : “<APG BASE>/bin/manage-modules.sh install trap-receiver”

Then, edit the file:

<APG BASE>/Event-Processing/Event-Processing-Manager/<INSTANCE>/conf/processing.xml

Add the following line:

```
<processing-element name="Trap-Receiver" type="Trap-Receiver" config="Trap-Receiver/Default/  
conf/config.xml" {STREAM_NAME}={STREAM_HANDLE} ... />
```

Here stream name is the name you will use to reference the string in your rule files (i.e. “data”). The handle refers to the destination for the stream name (i.e. “DW-A”). You can specify as many name, handle pairs as you desire, however the names must be unique.

The “config.xml” configuration file should be located in the “<APG BASE>/Event-Processing/Trap-Receiver/<INSTANCE>/conf/” directory. This file will contain your customized configuration for this processing element. Create and edit as explained in this document’s Configuration section.

Note: In order to run properly, this processing element may need to be run as root. You may safely skip to

the next section if you have no intention of running this module as root.

- Edit the “<APG BASE>/Event-Processing/Event-Processing-Manager/<INSTANCE>/conf/unix-service.properties” and uncomment
“# apg.user=root”
“# jvm.param.1=-Djava.rmi.server.hostname=\$HOSTNAME\$”
“# jvm.param.2=-Dcom.sun.management.config.file=conf/stats.properties”
- Regenerate the apg-event-processing-manager-* service by running “(sudo or su) <APG BASE>/bin/manage-modules.sh service update event-processing-manager <INSTANCE>”.
- Restart the APG event processing manager service by issuing “(sudo or su) <APG BASE>/bin/apg-event-processing-manager <INSTANCE> restart”.

Note: If the Trap-Receiver will need to be run as root, it is strongly recommended to do so using a distinct instance of the Event-Processing-Manager.

3

Configuration

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

1 Config example

The default configuration file for this collector is: <APG BASE>/Event-Processing/Trap-Receiver/<INSTANCE>/conf/conf.xml
Let's use the following example in order to create a configuration for this element:

```
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="http://www.watch4net.com/Events/TrapReceiver"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema">
  <rules value="rules" />
  <mibs value="mibs" />
  <ipbind incoming="0.0.0.0:10162" outgoing="0.0.0.0" />
  <threads count="2" />
  <incoming-v3-params username="" authentication-protocol="" authentication-password=""
  encryption-protocol="" encryption-password="" engineID="" />
```

```
<outgoing-v3-params username="" authentication-protocol="" authentication-password=""  
encryption-protocol="" encryption-password="" engineID="" />  
</config>
```

The basic provided configuration assumes the following:

- **<rules>** The rules file/folder that will be used. This is a mandatory element. Please note that you may only specify 1 rules tag.
- **<mibs>** The mibs file/folder that will be used. This is a mandatory element. When converting a trap to an event, the processing element will use the specified mib file(s) to convert OID's to human readable form. If the OID is not found, it will still be included as a raw OID. Please note that you may only specify 1 mibs tag.
- **<ipbind>** The interface that will be used to receive traps, as well as the interface that will be used to forward traps. These are optional parameters. If omitted, the default receiver is 0.0.0.0 port 162, and the default forwarder will forward using the first available interface. If you specify these parameters, you must specify the interface:port pairing, otherwise it will be invalid. If an invalid pairing is specified, or the element is unable to bind to the specified port, the program will terminate. IPv6 addresses are supported, using the format *[IPv6 address]:port*, as stated in RFC5952, section 6: example 1.
- **<threads>** The total number of threads for processing traps. If you are unsure of the optimal number of threads, you can safely omit this element and the module will configure it for you.
- **<incoming-v3-params>** This module will decode v3 traps using the information specified by this element. As all the attributes are blank in this case, this receiver will be unable to handle v3 traps. This element may be omitted. However, all attributes must be specified if the option is enabled. Encryption-Protocol is one of {"AES", "DES"}, Authentication-Protocol is one of {"MD5", "SHA"}. If the encryption-password is left blank, encryption will not be used. If the authentication-password is left blank, authentication will not be used. The engineid should be a string of ascii characters that correspond to the engineid of the agent. For instance, if your agent is using 0x70707071727374 as the engine, then the engineid should be "pppqrst".
- **<outgoing-v3-params>** This module will forward received v3 traps with the v3 parameters that are specified in this element. It can be configured in exactly the same way as the incoming-v3-params element.

Note: If the module is configured to run on a port numbered less than or equal to 1024, it will need to be run as root, otherwise it will terminate.

2 Rules configuration

The default rules folder will be in <APG-BASE>/Event-Processing/Trap-Receiver/<INSTANCE>/conf/rules/. Multiple rules files may be specified (by specifying the folder where they are located), and they will be processed in alphabetical order. Use the following rules file as a template for rule creation:

```
<?xml version="1.0" encoding="UTF-8"?>
<rules>
  <rule ip-address="*" oid="*" specific-id="*" generic-id="*" enterprise-id="*">
    <handle destination="data">
      <variablename-conversion from=".1.3.6.1.2.1.1.1.0" to="new_name" />
    </handle>
    <forward destination="192.168.1.100:45678" />
    <break />
  </rule>
  <rule ip-address="192.168.1.1/24" oid=".1.3.6.4.1.4.*" specific-id="2" generic-id="9">
    <handle destination="data" />
    <forward destination="10.128.2.32" />
    <break />
  </rule>
  <rule>
    <break />
  </rule>
</rules>
```

- <rule> This element accepts the following optional attributes, henceforth known as filters, ip-address, generic-id, specific-id, enterprise-id, oid. If these attributes are omitted then the rule will match everything.
 - <ip-address> This attribute may either be an ip address, or a netmask, as seen in the second example. It may also be a wildcard (*), however that is equivalent to omitting the attribute entirely. For v1 traps, this filtering is applied to the IP address of the sender, not of the agent.
 - <generic-id> This attribute may either be a number or a wildcard, however, as with the ip address, the wildcard is equivalent to omitting the attribute entirely. This attribute only applies to v1 traps.
 - <specific-id> This attribute may either be a number or a wildcard. This attribute only applies to v1 traps.
 - <oid> This attribute may either be an oid, or a wildcard, or the beginning of an oid terminated by a wildcard.

- `<enterprise-id>` This attribute may either be an oid, or a wildcard.
- `<forward>` This optional element will specify the desired destination of received traps that are matched by the filters of the associated rule. A destination, in the form of an ip address, must be specified, and a port may be specified (the default is 162). The module will forward all traps as traps to these destinations provided the filters are matched. IPv6 addresses are supported, using the format `[IPv6 address]:port`, as stated in RFC5952, section 6: example 1.
- `<handle>` This optional element will specify the stream to which the event generated from this trap is destined. This element also accepts conversion elements as children to specify data manipulations that can be performed/added to the event before it is forwarded on the stream.
- `<break>` This optional element will stop all further processing on the trap for all rules.

Note: The `generic-id`, `specific-id` and `enterprise-id` filter criteria also applies to SNMP v2 and v3 traps. Conversion to v1 traps, for filtering purposes only, is performed as per RFC 3584, section 3.2.

3 Conversion examples

As mentioned in the previous subsection, the `<handle>` element allows for the specifying of conversions that will be performed on the event data. The following is a comprehensive listing of those conversions, as well as examples of how each may be specified:

- `<constant-conversion>` This conversion is used to add a value to the event.

```
<constant-conversion key="key" value="value">
```

- `<variablename-conversion>` This conversion takes the key of a {key,value} pair from the generated event and converts it (This can be used to convert oids if they are not found in any of the specified mibs).

```
<variablename-conversion from="from" to="to">
```

4 A complete configuration example

This section provides a full, working and usable Trap-Receiver configuration example. The Event-Processing-Manager processing.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<processing-manager xmlns="http://www.watch4net.com/Events/DefaultProcessingManager" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.watch4net.com/Events/DefaultProcessingManager_
DefaultProcessingManager.xsd">
  <!-- Trap Receiver -->
  <processing-element enabled="true" name="Trap-Receiver" config="Trap-Receiver/Default/conf/
config.xml" data="EVENT-SPY" />

  <!-- Debugging -->
  <!-- Event spy for debugging purposes: outputs events to log files. Install the event-
processing-utils modules to use this. -->
  <processing-element name="EVENT-SPY" type="EventSpy" config="Event-Processing-Utils/Default
/conf" />
</processing-manager>
```

Note that *data* is used as the output stream name and that the *EVENT-SPY* component is the output stream target.

The Trap-Receiver config.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="http://www.watch4net.com/Events/TrapReceiver"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema">
  <rules location="rules" />
  <mibs location="mibs" />
  <ipbind incoming="0.0.0.0:10162" outgoing="0.0.0.0" />
</config>
```

Only SNMP v1 and v2c traps are handled by this configuration. SNMP v3 traps will be ignored.

MIB files provided in the conf/mibs directory are used to translate the trap's varbind OIDs and values to human-readable symbolic values.

IANAifType-MIB
IF-MIB
RFC1155-SMI
RFC1213-MIB
SNMPv2-CONF.mib
SNMPv2-MIB
SNMPv2-SMI
SNMPv2-TC

The Trap-Receiver conf/rules/rules00.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<rules xmlns="http://www.watch4net.com/Events/TrapReceiver"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema">
  <rule ip-address="10.128.2.1/24" generic-id="3" enterprise-id=".1.3.6.1.4.1.11970">
    <handle destination="data">
      <constant-conversion key="source" value="TRAP-RECEIVER" />
      <constant-conversion key="rule" value="subnet_2,_Watch4net_enterprise" />
      <variablename-conversion from=".1.3.6.1.4.1.11970.1.0" to="sent-by" />
    </handle>
  </rule>
</rules>
```

Sending this trap:

```
snmptrap -v 1 -c public 192.168.0.1:10162 .1.3.6.1.4.1.11970 192.168.0.2 3 0 ''
.1.3.6.1.2.1.2.2.1.2.2 s 'GigabitEthernet0/1' .1.3.6.1.2.1.2.2.1.3.2 i 6
.1.3.6.1.4.1.11970.1.0 s 'Watch4Net APG Alerting'
```

```
Snmp version:      1
Community string:  public
To:                192.168.0.1:10162
Enterprise-id:     .1.3.6.1.4.1.11970
Agent:             192.168.0.2
Trap generic id:   3
Trap specific id:  0
Timestamp:         Current sysuptime
.1.3.6.1.2.1.2.2.1.2.2 s 'GigabitEthernet0/1' (ifDescr.2)
.1.3.6.1.2.1.2.2.1.3.2 i 6 (ifType.2)
.1.3.6.1.4.1.11970.1.0 s 'Watch4Net APG Alerting'
```

Will generate this event:

mib	= IF-MIB	(OCCURRENCE,STRING)
ifDescr.2	= GigabitEthernet0/1	(OCCURRENCE,STRING)
ifType.2	= ethernetCsmac	(OCCURRENCE,STRING)
source	= TRAP-RECEIVER	(OCCURRENCE,STRING)
rule	= linkdown on subnet 2, Watch4net	(OCCURRENCE,STRING)
sent-by	= Watch4Net APG Alerting	(OCCURRENCE,STRING)
sourceIP	= 192.168.0.1	(OCCURRENCE,STRING)
generic	= 3	(OCCURRENCE,INT)
specific	= 0	(OCCURRENCE,INT)
enterprise	= .1.3.6.1.4.1.11970	(OCCURRENCE,STRING)
timestamp	= 1338317765605	(OCCURRENCE, LONG)
community	= public	(OCCURRENCE,STRING)
sysUpTime	= 53952559	(OCCURRENCE, LONG)
agent	= 192.168.0.2	(OCCURRENCE,STRING)

5 Reconfiguration

Should the module need to be reconfigured while it is running, simply make the desired modifications, and then invoke the reload-trap-receiver-config script (.sh or .cmd) contained in the <APG-BASE>/bin directory. It is important to note that for this reconfiguration to take place, the webservice-gateway component must be running. Additionally, if the webservice-gateway install has been customized, the changes will need to be reflected in the <APG-BASE>/Event-Processing/Trap-Receiver/<INSTANCE>/conf/ws.properties file.

```
url=https://localhost:48443/Event-Processing/Event-Processing-Manager/Default/TRAP-RECEIVER
username=admin
password=changeme
sslvalidation=false
```

- <url> This is the url for the webservice gateway. It may need to be changed if you have not used the default install. If you have disabled ssl validation, you will most likely need to change the https for http. If your webservice-gateway task has been customized, you may need to extract the host:port from it's properties file. This can be found in <APG-BASE>/Tools/Webservice-Gateway/<INSTANCE>/conf/module.properties. You will then need to replace the localhost:48443 with the information found in that file. If you have created an instance of the trap-receiver that is not the default, you will have to replace the Default in the url with the name of the instance, and finally, if you have changed the name of the processing element (found in <APG-BASE>/Event-Processing/Event-Processing-Manager/<INSTANCE>/conf/processing.xml), you will need to change the trap-receiver to the name found there.
- <username> This is the username for the webservice-gateway. It can be found in <APG-BASE>/Tools/Webservice-Gateway/<INSTANCE>/conf/module.properties.
- <password> This is the password for the webservice-gateway. It can be found in <APG-BASE>/Tools/Webservice-Gateway/<INSTANCE>/conf/module.properties.
- <sslvalidation> This option lets the tool know whether or not you would like to use ssl validation when making your web request. Acceptable values are true or false.

Note: Additionally, these options may be specified with command line arguments that will override any information contained in the properties file.

There are two ways in which this reload may take place. Either the rules in the current rules directory may be changed, or the rules location contained in <APG-BASE>/Event-Processing/Trap-

Receiver/<INSTANCE>/conf/config.xml may be changed.

Note: This reconfiguration applies only to rules. No other component of the module may be reloaded while the module is still running.
