

Create remote event

- [Introduction](#)
- [Considerations](#)
 - [Sending UP events to the Primary](#)
 - [Reorder protection](#)
 - [Poller sending duplicate events to the primary](#)
- [Fast create remote event](#)
 - [Debug Values](#)
 - [Boolean Values](#)
 - [Retries](#)
 - [Version History](#)

Introduction

Note: This guide is intended to be for opHA 3, opEvents 3 and NMIS 9. The tool is available for previous versions, but the syntax and some parameters can change.

In order to integrate the poller events into the primary server on an opHA environment, we can make use of the `create_remote_event.pl` tool.

`/usr/local/omk/bin/create_remote_event.pl`

```
Usage: create_remote_event.pl -s <base_url> -u <user> -p <passwd> [-i] <no_ssl_validation> [-q]
{eventprop=value...| < json file>
```

```
base_url: http://servername:portnumber/omk, API endpoint details are automatic.
-q: quiet, don't print the new event's id on STDOUT
optional, -i: No arguments, use SSL without validation (self-signed certs).
```

```
you must either provide all required event properties as name=value pairs
or as JSON hash on STDIN.
```

We can then create some event actions rules to send the poller events the the primary server:

1. Edit the Event Actions (In opEvents, navigate to System, then Edit Event Actions) and create a new script (replacing the below with your configurations):
 - a. PRIMARY-URL (this is the Primary server the event will be sent to)
 - b. USERNAME (A user account on the Primary server with Admin rights)
 - c. PASSWORD (the password for the USERNAME account)
 - d. Authority (this is the button label that will appear on the Primary)
 - e. POLLER-URL (this is the Poller server sending the event, used to link back to the originating event from the Primary server)

```
"opevents_primary" : {
  "exec" : [ "/usr/local/omk/bin/create_remote_event.pl", "-s", "http://PRIMARY-URL.
opmantek.net/omk", "-u", "USERNAME", "-p", "PASSWORD", "authority='POLLER HUMAN NAME' " ],
  "arguments" : [ "location=http://POLLER-URL.opmantek.com/en/omk/opEvents/events/event.
_id/event_context", "node=node.name", "event=event.event", "details=event.details", "time=event.
time", "date=event.date", "element=event.element", "interface_description=event.
interface_description", "type=event.type", "priority=event.priority", "level=event.level",
"nodeType=node.nodeType", "state=event.state", "stateful=event.stateful" ],
  "output" : "save",
  "stderr" : "save",
  "exitcode" : "save"
}
```

Please note, you can edit the event information that you want to send.

2. Add a new policy in the Event Actions file:

```
"100" : {
  "IF" : "event.priority >= 1",
  "THEN" : "script.opevents_primary()",
  "BREAK" : "true"
}
```

3. Test the remote event:

```
/usr/local/omk/bin/create_remote_event.pl -s http://primary.opmantek.net/omk -u nmis -p password
authority='poller nine Poller' location=http://primary.opmantek.com/en/omk/opEvents/events
/600161b500eff2351645be2c/event_context host= event="Node Configuration Change" details="Changed at 140 days 1:
00:55" node="bnelab-rr1" time=1610703283 date= element= interface_description= type=nmis_eventlog priority=3
level=Warning nodeType= state= stateful=
60016d06891ad2506c49ad72
```

The raw exit code should be 0:

```
[Mon Jan 18 07:31:50 2021] [info] worker.action[6595] 6005396300eff20fc35d55eb method='opevents_primary' got
raw exitcode=0 from pid=6602 exec=['/usr/local/omk/bin/create_remote_event.pl', '-s', 'http://primary.opmantek.net
/omk', '-u', 'nmis', '-p', 'password', 'authority=\'poller nine Poller\'', 'location=http://primary.opmantek.com/en
/omk/opEvents/events/6005396300eff20fc35d55eb/event_context', 'node=rbogon344', 'event=Node Configuration
Change', 'details=Changed at 142 days 23:00:
55', 'time=1610955103', 'date=', 'element=', 'interface_description=', 'type=nmis_eventlog', 'priority=3', 'level=Warni
ng', 'nodeType=', 'state=', 'stateful=']
```

Considerations

Sending UP events to the Primary

When an event is acknowledged, opEvents stops the propagation. That means, that all the rules are not going to run anymore. There are different approaches to achieve this, but is a common schema to set up **opevents_auto_acknowledge_up** to false in the poller, so the event is not acknowledge and can be sent to the primary.

More information: [https://community.opmantek.com/pages/viewpage.action?pageId=27264053#EventActionsandEscalation\(opEvents3\)-ActionPolicyApplicationandTiming](https://community.opmantek.com/pages/viewpage.action?pageId=27264053#EventActionsandEscalation(opEvents3)-ActionPolicyApplicationandTiming)

Reorder protection

Forwarded events may arrive out of order to the primary server, because of network congestion or slow action processing.

To enable reorder protection, two steps need to be taken:

- Set the configuration property **state_reorder_window** to a positive number (e.g. 30) on the receiving server.
- Always send the **authority** property, to denote the event as originating from a remote authoritative source.

More information: [Deduplication and storm control in opEvents#StatefulDeduplication,ForwardedEventsandReorderProtectionreorder_protection](#)

Poller sending duplicate events to the primary

If your poller is sending duplicate events to the primary make sure that your EventActions.json script does not have any 'or' gates in your 'IF' statements - 'AND' and 'and' are fully supported, but 'or' (also 'OR') is not supported and can lead to unexpected behavior.

It's also best practice to use "BREAK" : "true" whenever possible, for example:

If we confirm that event.event = 'Node Down' there's no need to also check to see if event.event = 'Interface Down' etc.

Fast create remote event

We have recently rewritten create remote event in Go and have provided the option to use token auth skipping out one of the auth requests from the old Perl version.

We recommend you place this in the current location of create_remote_event.pl which is in /usr/local/omk/bin

We have retained all arguments from the perl version.

Version 1.1.0 can be downloaded here: https://dl-omk.opmantek.com/remote_event/fast-remote-event-1.1.0-Linux-x86_64.bin

```

-d int          Log Level (default 4)
-p string       Password for the remote user
-path string    Path for generate auth token script (default "/usr/local/omk/bin/generate_auth_token.pl")
-q             Quiet level, (1|true|0|false). Don't output anything (default true), -q=0 to see event id of
remote system.
-retry int      Maximum number of retries
-s string       http://servername:portnumber/omk, API endpoint details are automatic. (default "http://localhost:
8042/omk")
-t string       Token for auth
-u string       Username for remote server (default "nmis")
-v             Verify the servers TLS connection (default true)
-version        Print out the version

```

New in this version is using token auth so you can now skip passing the username and password and use a token from the master.

More about tokens found here in [Delegated Authentication](#)

This is passed using -t

```

fast-remote-event-1.0.1-LinuxX86_64.bin -s https://primary-one.opmantek.com/omk -t myexampletoken
event=testevent host=localhost

```

If your Opmantek Applications in a non standard location you can use -path to point to the full path of generate_auth_token.pl which is shipped in /usr/local/omk/bin/generate_auth_token.pl in future versions we aim to remove this.

Debug Values

to see debug set value to one i.e. -d 1, debug comes in as stdout (standard output)

Boolean Values

for the TLS verify command we support the following flags

1, 0, t, f, T, F, true, false, TRUE, FALSE, True, False

Retries

Sometimes the end server is busy but using the flag -retry you can instruct the http handler to make X amount of requests to try again, this will perform a back off to try and not swamp the end server.

Version History

V1.1.0

8th June 2023

Fix issue where fast-remote-event would crash when receiving a a server response other than JSON.

V1.0.1

Internal Release

V1.0.0

First Release

