

Integrating IBM Spectrum Scale with external monitoring systems

Anna Greim & Norbert Schuld

Question:

Can I extend the health events with my own?

Answer:

Yes, you can! And here is how....

Step 1:

Create the custom.json file

Step 2:

Make system aware of your new events

Step 3:

Raise your custom events

Step 1:

Create the custom.json file

Step 2:

Make system aware of your new events

Step 3:

Raise your custom events

Create json in update-save path:

```
# touch /var/mmfs/mmsysmon/custom.json
```

Create json in update-save path:

```
# touch /var/mmfs/mmsysmon/custom.json
```

Create json in update-save path:

```
# touch /var/mmfs/mmsysmon/custom.json
```


Create symlink to active path:

```
# ln -s /var/mmfs/mmsysmon/custom.json  
/usr/lpp/mmfs/lib/mmsysmon/custom.json
```

Create symlink to active path:

```
# ln -s /var/mmfs/mmsysmon/custom.json  
/usr/lpp/mmfs/lib/mmsysmon/custom.json
```

Create **symlink** to active path:

```
# ln -s /var/mmfs/mmsysmon/custom.json  
/usr/lpp/mmfs/lib/mmsysmon/custom.json
```

Create symlink to active path:

```
# ln -s /var/mmfs/mmsysmon/custom.json  
/usr/lpp/mmfs/lib/mmsysmon/custom.json
```

Check current number of events:

```
# cd /usr/lpp/mmfs/lib/mmsysmon/
```

```
# ./RASEvent.py
```

...

```
number of events 752
```

Check current number of events:

```
# cd /usr/lpp/mmfs/lib/mmsysmon/
```

```
# ./RASEvent.py
```

...

```
number of events 752
```

Check current number of events:

```
# cd /usr/lpp/mmfs/lib/mmsysmon/
```

```
# ./RASEvent.py
```

...

```
number of events 752
```

Check current **number of events**:

```
# cd /usr/lpp/mmfs/lib/mmsysmon/  
# ./RASEvent.py
```

...

number of events 752

Write event in json format into custom.json:

```
# vi /var/mmfs/mmsysmon/custom.json
```

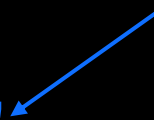
Write event in json format into custom.json:
vi /var/mmfs/mmsysmon/custom.json

```
{  
  "event_1": {  
    "cause": "",  
    "code": "cu_001",  
    "event_type": "INFO",  
    "user_action": "",  
    "scope": "NODE",  
    "description": "",  
    "severity": "INFO",  
    "message": "日本 arg1{0}, arg2{1}"  
  }  
}
```

```
{  
  "event_1": { ← unique name  
    "cause": "",  
    "code": "cu_001",  
    "event_type": "INFO",  
    "user_action": "",  
    "scope": "NODE",  
    "description": "",  
    "severity": "INFO",  
    "message": "日本 arg1{0}, arg2{1}"  
  }  
}
```


```
{
  "event_1": {
    "cause": "",
    "code": "cu_001",
    "event_type": "INFO",
    "user_action": "",
    "scope": "NODE",
    "description": "",
    "severity": "INFO",
    "message": "日本 arg1{0}, arg2{1}"
  }
}
```

unique code,
max 6 chars



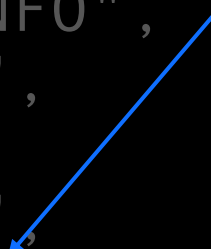
```
{
  "event_1": {
    "cause": "",
    "code": "cu_001",
    "event_type": "INFO",
    "user_action": "",
    "scope": "NODE",
    "description": "",
    "severity": "INFO",
    "message": "日本 arg1{0}, arg2{1}"
  }
}
```

always INFO




```
{
  "event_1": {
    "cause": "",
    "code": "cu_001",
    "event_type": "INFO",
    "user_action": "",
    "scope": "NODE",
    "description": "",
    "severity": "INFO",
    "message": "日本 arg1 {0}, arg2 {1}"
  }
}
```

INFO,
WARNING or
ERROR



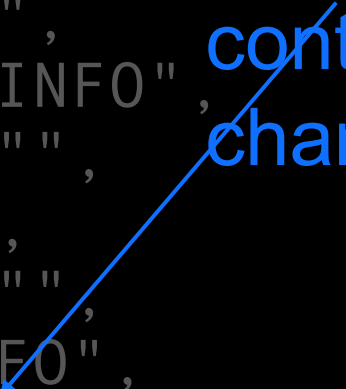
```
{
  "event_1": {
    "cause": "",
    "code": "cu_001",
    "event_type": "INFO",
    "user_action": "",
    "scope": "NODE",
    "description": "",
    "severity": "INFO",
    "message": "日本 arg1 {0}, arg2 {1}"
  }
}
```

placeholders
for args




```
{
  "event_1": {
    "cause": "",
    "code": "cu_001",
    "event_type": "INFO",
    "user_action": "",
    "scope": "NODE",
    "description": "",
    "severity": "INFO",
    "message": "日本 arg1 {0}, arg2 {1}"
  }
}
```

unicode
possible, no
control
characters



Step 1:

Create the custom.json file

Step 2:

Make system aware of your new events

Step 3:

Raise your custom events

Restart GUI service:

```
# systemctl restart gpfsgui.service
```

Restart GUI service:

```
# systemctl restart gpfsgui.service
```

Restart mmsysmon service (version < 5.0.2-1):
mmsysmoncontrol restart

Restart mmsysmon service (version < 5.0.2-1):
mmsysmoncontrol restart

Restart mmsysmon service (version \geq 5.0.2-1):
`# systemctl restart mmsysmon.service`

Restart mmsysmon service (version $\geq 5.0.2-1$):
`# systemctl restart mmsysmon.service`

Check current **number of events**:

```
# cd /usr/lpp/mmfs/lib/mmsysmon/  
# ./RASEvent.py
```

...

```
number of events 753
```

Step 1:

Create the custom.json file

Step 2:

Make system aware of your new events

Step 3:

Raise your custom events

Issue following command to raise events:

```
# mmsysmonc event custom ev_name args
```

Issue following command to raise events:

```
# mmsysmonc event custom ev_name args
```

Example:

```
# mmsysmonc event custom event_1 a,b  
Event event_1 raised
```

Example:

```
# mmsysmonc event custom event_1 a,b  
Event event_1 raised
```

Example:

```
# mmsysmonc event custom event_1 a,b  
Event event_1 raised
```

To be found on CLI:

```
# mmhealth node eventlog
```

```
...
```

```
... event_1 INFO 日本 arg1 a, arg2 b
```


To be found on CLI:

```
# mmhealth node eventlog
```

...

```
... event_1 INFO 日本 arg1 a, arg2 b
```

event name

To be found on CLI:

```
# mmhealth node eventlog
```

...

```
... event_1 INFO 日本 arg1 a, arg2 b
```



Severity

To be found on CLI:

```
# mmhealth node eventlog
```

...

```
... event_1 INFO 日本 arg1 a, arg2 b
```



message

To be found on CLI:

```
# mmhealth node eventlog
```

...

```
... event_1 INFO 日本 arg1 a, arg2 b
```

arguments
included in
message

To be found on CLI:

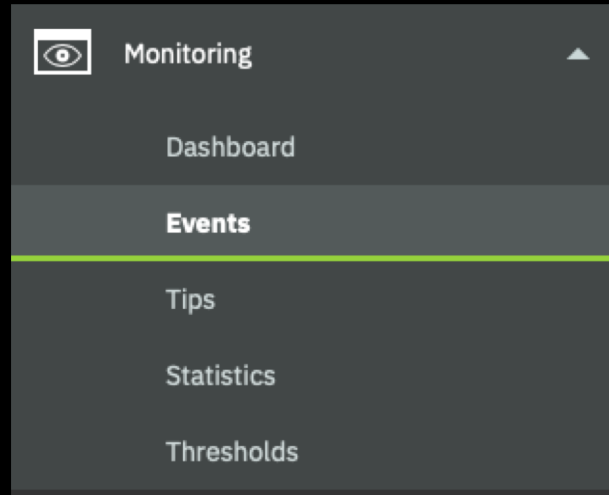
```
# mmhealth node eventlog
```

```
...
```

```
... event_1 INFO 日本 arg1 a, arg2 b
```



To be found on GUI:



Event Groups

Individual Events

View Details

Actions

Notices

↺

Last Updated: 11:08

↓

Export

Severity	Event Time	Reporting N...	Event Name	↑	Event T...	Entity Name	Occurren...	Message	Component
<div><div>Info</div></div>	05.03.19 11:07:57	spectrumscale	event_1		Notice	spectrumscale	1	日本 arg1 a, arg2 b	Custom Events
<div><div>Info</div></div>	05.03.19 11:07:50	spectrumscale	eventlog_cleared		Notice	spectrumscale	1	On the node spectrumscale the eventlog was cleared.	Cluster State

Event Groups

Event Groups

Individual Events

View Details

Actions

Notices

Last Updated: 11:08

Export

Severity	Event Time	Reporting N...	Event Name	Event T...	Entity Name	Occurren...	Message	Component
Info	05.03.19 11:07:57	spectrumscale	event_1	Notice	spectrumscale	1	日本 arg1 a, arg2 b	Custom Events
Info	05.03.19 11:07:50	spectrumscale	eventlog_cleared	Notice	spectrumscale	1	On the node spectrumscale the eventlog was cleared.	Cluster State

Notices ▾

Event Groups									
Individual Events									
<div>View Details Actions ▾ Notices ▾ Last Updated: 11:08 Export</div>									
Severity	Event Time	Reporting N...	Event Name	Event T...	Entity Name	Occurren...	Message	Component	
Info	05.03.19 11:07:57	spectrumscale	event_1	Notice	spectrumscale	1	日本 arg1 a, arg2 b	Custom Events	
Info	05.03.19 11:07:50	spectrumscale	eventlog_cleared	Notice	spectrumscale	1	On the node spectrumscale the eventlog was cleared.	Cluster State	

Event Name



event_1

Event Groups									
Individual Events									
<div>View Details Actions Notices Last Updated: 11:08 Export</div>									
Severity	Event Time	Reporting N...	Event Name	Event T...	Entity Name	Occurren...	Message	Component	
Info	05.03.19 11:07:57	spectrumscale	event_1	Notice	spectrumscale	1	日本 arg1 a, arg2 b	Custom Events	
Info	05.03.19 11:07:50	spectrumscale	eventlog_cleared	Notice	spectrumscale	1	On the node spectrumscale the eventlog was cleared.	Cluster State	

Message

日本 arg1 a, arg2 b

Event Groups

Individual Events

View Details

Actions

Notices

Last Updated: 11:08

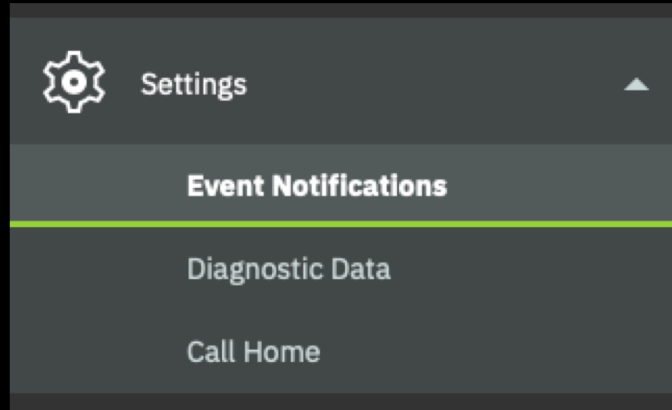
Export

Severity	Event Time	Reporting N...	Event Name	Event T...	Entity Name	Occurren...	Message	Component
Info	05.03.19 11:07:57	spectrumscale	event_1	Notice	spectrumscale	1	日本 arg1 a, arg2 b	Custom Events
Info	05.03.19 11:07:50	spectrumscale	eventlog_cleared	Notice	spectrumscale	1	On the node spectrumscale the eventlog was cleared.	Cluster State

Component
Custom Events

Event Groups									
Individual Events									
<div> View Details Actions Notices ↻ Last Updated: 11:08 Export </div>									
Severity	Event Time	Reporting N...	Event Name	Event T...	Entity Name	Occurren...	Message	Component	
Info	05.03.19 11:07:57	spectrumscale	event_1	Notice	spectrumscale	1	日本 arg1 a, arg2 b	Custom Events	
Info	05.03.19 11:07:50	spectrumscale	eventlog_cleared	Notice	spectrumscale	1	On the node spectrumscale the eventlog was cleared.	Cluster State	

Usable with E-Mail Notification in GUI:



Event Notifications


Email Server

Email Recipients

SNMP Manager

 Create Recipient

Actions 

 Refresh

 Export

Event Notifications


Email Server

Email Recipients

SNMP Manager

 Create Recipient




Actions 

 Refresh


 Export

Create Recipient

Name:


  

Email address:




Severity


Warning



Events



Report



☐ Custom Events

☐ Object Authentication

☐ CES

☐ CES ID

Cancel

Create

Create Recipient

Name:

!

Email address:

*

Severity

Warning

▼

Events

▼

Report

▼

☐ Custom Events

☐ Object Authentication

☐ CES

☐ CES ID

Cancel

Create

Report

☐ Custom Events

Questions?

Question:

Can I export the health events into my own tool?

Answer:

Don't doubt us! Of course you can

Option 1. REST API
Option 2. Callout script

Option 1. REST API
Option 2. Callout script

Getting health states:

```
curl -k -u <gui_user>:<password> -X GET -H  
content-type:application/json  
"https://<gui_node>:443/scalemgmt/v2/nodes/<node_name>/health/states"
```

Getting health states:

```
curl -k -u <gui_user>:<password> -X GET -H  
content-type:application/json  
"https://<gui_node>:443/scalemgmt/v2/nodes/<node_name>/health/states"
```


Getting health states:

```
curl -k -u <gui_user>:<password> -X GET -H  
content-type:application/json  
"https://<gui_node>:443/scalemgmt/v2/nodes/<node_name>/health/states"
```

Getting **health states**:

```
curl -k -u <gui_user>:<password> -X GET -H  
content-type:application/json  
"https://<gui_node>:443/scalemgmt/v2/nodes/<node_name>/health/states"
```

Getting health events:

```
curl -k -u <gui_user>:<password> -X GET -H  
content-type:application/json  
"https://<gui_node>:443/scalemgmt/v2/nodes/<node_name>/health/events"
```

Getting **health events**:

```
curl -k -u <gui_user>:<password> -X GET -H  
content-type:application/json  
"https://<gui_node>:443/scalemgmt/v2/nodes/<node_name>/health/events"
```

Option 1. REST API
Option 2. Callout script

Callout Script:

1. be root
2. create /var/mmfs/etc/eventsCallback
3. make it executable
4. wait for events to be raised

```
1. be root:  
# whoami  
root
```

```
1. be root:  
# whoami  
root
```


2. create /var/mmfs/etc/eventsCallback

```
# touch /var/mmfs/etc/eventsCallback
```

```
2. create /var/mmfs/etc/eventsCallback  
# touch /var/mmfs/etc/eventsCallback
```

3. make it executable:

```
# cd /var/mmfs/etc/  
# chmod +x eventsCallback
```

3. make it executable:

```
# cd /var/mmfs/etc/
```

```
# chmod +x eventsCallback
```

3. make it executable:

```
# cd /var/mmfs/etc/
```

```
# chmod +x eventsCallback
```

Call arguments:

version, date, time, timezone, event, component,
identifier, severity, state, message,
event_arguments

Call arguments:

version, date, time, timezone, event, component,
identifier, severity, state, message,
event_arguments

integer

Call arguments:

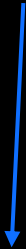
version, date, time, timezone, event, component,
identifier, severity, state, message,
event_arguments

yyyy-mm-dd

Call arguments:

version, **date**, time, timezone, event, component,
identifier, severity, state, message,
event_arguments

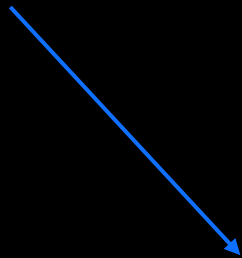
hh:MM:ss.mmm



Call arguments:

version, date, **time**, timezone, event, component,
identifier, severity, state, message,
event_arguments

e.g. CET



Call arguments:

version, date, time, **timezone**, event, component,
identifier, severity, state, message,
event_arguments

/usr/lpp/mmfs/lib/mmsysmon/*.json



Call arguments:

version, date, time, timezone, **event**, component,
identifier, severity, state, message,
event_arguments

e.g. GPFS, FILESYSTEM, etc.



Call arguments:

version, date, time, timezone, event, component,
identifier, severity, state, message,
event_arguments

e.g. fs1, gpfs0, “

Call arguments:

version, date, time, timezone, event, component,
identifier, severity, state, message,
event_arguments


first letter of INFO, TIP, WARNING,
ERROR

Call arguments:

version, date, time, timezone, event, component,
identifier, severity, state, message,
event_arguments

first letter of HEALTHY, TIPS,
DEGRADED, ERROR or X

Call arguments:

version, date, time, timezone, event, component,
identifier, severity, state, message,
event_arguments

including arguments, in “



Call arguments:

version, date, time, timezone, event, component,
identifier, severity, state, message,
event_arguments

comma separated, in “, %-encoded

Call arguments:

version, date, time, timezone, event, component,
identifier, severity, state, message,
event_arguments

Example:

```
# vi /var/mmfs/etc/eventsCallback
```

Example:

```
# vi /var/mmfs/etc/eventsCallback
```

```
#!/usr/bin/bash  
echo $* >> ~/eventsCallback.log
```

```
# mmsysmonc event custom event_1 a,b
```

```
# vi ~/eventsCallback.log
```

```
1 2019-03-07 12:51:44.342376 EST  
event_1 custom I X 日本 arg1 a, arg2 b  
a,b
```


Questions?

Custom Events:

https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.2/com.ibm.spectrum.scale.v5r02.doc/bl1adv_createuserdefinedevents.htm

REST API health states:

[https://www.ibm.com/support/knowledgecenter/en/
STXKQY_5.0.2/com.ibm.spectrum.scale.v5r02.doc/
bl1adm_apiv2getnodehealthstates.htm](https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.2/com.ibm.spectrum.scale.v5r02.doc/bl1adm_apiv2getnodehealthstates.htm)

REST API health events:

[https://www.ibm.com/support/knowledgecenter/en/
STXKQY_5.0.2/com.ibm.spectrum.scale.v5r02.doc/
bl1adm_apiv2getnodehealthevents.htm](https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.2/com.ibm.spectrum.scale.v5r02.doc/bl1adm_apiv2getnodehealthevents.htm)

Callback Script:

https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.2/com.ibm.spectrum.scale.v5r02.doc/bl1adv_createscriptforevents.htm

Thank You!

