

## Dollar Universe and Sysload integration

Follow this how to procedure to trigger a Dollar Universe task based on a Sysload Alert

### Define the Sysload Alert to monitor

First, you need to configure your Alert Manager to tell him which alerts from which Sysload agents you want to monitor. Follow the procedure:

- Open the Alert Manager installation directory ("C:\Program Files\sysload\slald" by default).
- Edit the "Agents.txt" file.
- Add the definition of the agent you want to monitor by the line:

```
<Agent>{TYPE}:{HOST}:{INSTANCE}</Agent>
```

For example, to monitor "my\_host" windows agent:

```
<Agent>NT:my_host:*</Agent>
```

If you want to trigger monitor several Sysload Agents at the same time, you just need to add as much line as described above.

```
<Agent>NT:my_host:*</Agent>
[CODE]<Agent>UNIX:unix_host:*</Agent>[/CODE]
```

[u]NB: [/u] Do not forget to update the name of the Manager Server or to add a new <[i]Group[/i]> section if need be.[list]  
[\*]Edit the [i]"Reactions.txt[/i]" file.  
[\*]Add a new <[i]reaction[/i]> section as follow:

```
[CODE]
<reaction>
<filters>
<filter>
<agents>
{MGTSERVER}:{TYPE}:{HOST}:{INSTANCE}
</agents>
<alertname>{ALERTNAME}</alertname>
<alertpriority>{ALERTPRIORITY}</alertpriority>
<alertstate>{ALERTSTATE}</alertstate>
</filter>
</filters>
<actions>
<cmd><![CDATA[cscript {PATH_TO_YOUR_SCRIPT} "%c" "%a" "%ai" "%t" "%x" "%p"
"%s" "%d" "%i" "%m"]]></cmd>
</actions>
</reaction>
[/CODE]
```

Where {MGTSERVER}, {TYPE}, {HOST} and {INSTANCE} give the definition of the monitored agent.

Where {ALERTNAME}, {ALERTPRIORITY} and {ALERTSTATE} give the definition of the monitored alert.

The possible states for the Sysload alerts are "begin", "end" and "persist".

{PATH\_TO\_YOUR\_SCRIPT} indicates the path to the given script that will launch the \$U trigger.

If you want to trigger several Sysload alerts, you could add them under <alertname> or <alertstate> section as follow:

```
[CODE]
<alertname>{ALERTNAME1},{ALERTNAME2}</alertname>
<alertpriority>{ALERTPRIORITY1},{ALERTPRIORITY2}</alertpriority>
<alertstate>{ALERTSTATE1},{ALERTSTATE2}</alertstate>
[/CODE]
```

Or you could add a new <[i]reaction[/i]> section as described above.

[\*]Restart your Alert Manager.

[/list]

[size=5][b]Define which \$U node to target[/b][/size]

[list]

[\*]Edit the given script

[\*]Inform the attributes giving the definition of the target \$U node:

[list]

[\*][i][b>host[/i]: [i]The hostname of the \$U node.

[\*][i][b>port[/i]: [i]The port number of the \$U api.

[\*][i][b>area[/i]: The target area.

[/list][\*]Inform the attributes giving the way you are going to authenticate yourself to \$U node:

[list]

[\*][i][b>authentication\_key[/i]: The authentication key you got via UVC.

or

[\*][i][b>user[/i] [i]/ [b>password[/i]: Your credentials.

[/list][u]NB:[/u] You must inform either the authentication key or your credentials.

[u]NB:[/u] If you inform both the authentication key and your credentials, only the authentication will be taken into account.[\*][optional] You can modify the event type that will be raised on \$U. By default this event type is: "[b>SYSLOAD[/b]".

[\*]Save and close the script.

[/list]

[size=5][b>Event properties[/b][/size]

The given script consider few alert related properties by default that are:

[list]

[\*][i][b>AGENT\_TYPE[/i]: The type of the agent.

[\*][i][b>AGENT\_IP[/i]: The IP address of the agent.

[\*][i][b>AGENT\_INSTANCE[/i]: The instance of the agent.

[\*][i][b>ALERT\_NAME[/i]: The name of the alert.

[\*][i][b>ALERT\_INSTANCE[/i]: The instance of the alert.

[\*][i][b>ALERT\_PRIORITY[/i]: The priority of the alert.

[\*][i][b>ALERT\_STATE[/i]: The state of the alert ("begin", "end" or "persist").

[\*][i][b>ALERT\_DATE[/i]: The date at which the alert was triggered.

[\*][i][b>ALERT\_DURATION[/i]: The duration of the alert in number of iterations of the Sysload Collector.

[\*][i][b>ALERT\_MESSAGE[/i]: The alert message.

[/list]

If you want to modify this list of properties, you need to edit the [i]<cmd>[/i] attribute in the "[i]Reactions.txt[/i]" file of your Alert Manager and the given script.

[b][size=5]Output[/size][/b]

[color=#333333][font=Arial, Helvetica, sans-serif][size=3]Basically, the output of the script will be something like:[/size][/font][/color]

[CODE]

Script launched the YYYY/MM/DD at hh:mm:ss

Login on {HOST}:{PORT} -> Success

Launch the TEST event -> Incomplete

=> Trigger: TEST1 -> Launch number: XXXXXXXX

=> Trigger: TEST2 -> Error [1023]: Only provoked tasks can be triggered.

Logout -> Success

Then the output gives you basic trigger related operations:

- Login (if no authentication key given)
- Event type launch
- Logout (if no authentication key given)

It will give you the launch number of the launched jobs, or the code and error message if a launch has failed.

**NB:** By default this is logged into a .log file with the same name as your script. You can transform it to a console output by modifying the ***log\_to\_file*** attribute value to "false".