Prerequisites

- Unix with **Syslog-ng** server for system logging.
- cURL (cf. http://curl.haxx.se/)

Define the Syslog event to monitor

- Go to directory containing your "*syslog-ng.conf*" file (this should be /etc/syslog-ng).
- Edit "*syslog-ng.conf*" file.
- Go at the end of the "***Destinations***" pseudo-section.
- Add the following line, that defines a template output:

```
$template t_orsyp_trigger {
template("$PROGRAM|$HOST|$FACILITY_NUM|$LEVEL_NUM|$DATE|$MSG"); };
```

- Add the following line, that defines a new destination for Syslog events:

```
destination d_orsyp_trigger { program("ksh {PATH_TO_YOUR_SCRIPT"}
template(t_orsyp_trigger)); };
```

Where {**PATH_TO_YOUR_SCRIPT**} indicates the path to the given script that will launch the $U trigger.

- Go at the end of the "***Log paths***" pseudo-section.
- Add the following line, that defines which kind of events we want to monitor:

```
log { source({YOUR_SOURCE}); filter({YOUR_FILTER});
destination(d_orsyp_trigger); };
```

Where {***YOUR_SOURCE***} defines the considered source of event and {***YOUR_FILTER***} the filter on the Syslog events you want to trigger.
You have predefined source and filter defined into "***Sources***" and "***Filters***" pseudo-section of the same "syslog-ng.conf" file.
If you want to trigger several Syslog events, you can add other lines of that kind. We could have for example, something like:

```
log { source(s_src); filter(f_kern); destination(d_orsyp_trigger); };
log { source(s_src); filter(f_auth); destination(d_orsyp_trigger); };
log { source(s_src); filter(f_crit); destination(d_orsyp_trigger); };
```

Please refer to the **Syslog-ng** documentation for filter definition (cf. http://www.balabit.c...ce_filters.html)

Define which $U node to target

- Inform the attributes giving the definition of the target $U node:

    - ***host:*** The hostname of the $U node.
    - ***port:*** The port number of the $U api.
    - ***area***: The target area.
- Inform the attributes giving the way you are going to authenticate yourself to $U node:

- ***authentication_key***: The authentication key you got via UVC OR
- ***user*** / ***password***: Your credentials.

NB: You must inform either the authentication key or your credentials.
NB: 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: "**SYSLOG_EVENT**".
- Save and close the script.

Ensure that the script is executable by the **syslog-ng** server.
NB: As the given script is persistent, you will have to restart **syslog_ng** service every time you modify this script ; to update the authentication key for example.

Event properties

By default, the script provided transmits the following event properties:

- ***PROGRAM***: The name of the program that raised the Syslog event.
- ***HOST***: The host of this program.
- ***FACILITY***: The Syslog facility level numeric value (cf. http://en.wikipedia....Facility_levels).
- ***SEVERITY***: The Syslog severity level numeric value (cf. http://en.wikipedia....Severity_levels).
- ***DATE***: The date/time when the Syslog event has been raised.
- ***MESSAGE***: The message of the Syslog event.

Customize the transmitted event properties

If you want to add or remove some event properties, you should modify the ***t_orsyp_trigger*** template or add a new one.
The template have to be like this: "*$PROPERTY1|$PROPERTY2|...|$PROPERTYN*"
With '|' separating each considered property.
Please refer to this page for the list of the available properties: http://www.balabit.c...nce-macros.html.
Then, you also need to modify the given script accordingly.
Just search for the following comment to find the places where you have to modify the script:

```
# You should modify these lines if you modify the list of considered event
properties
```

Output

Basically, the output of the script will be something like:

```
Script launched at {DATE}
Login on {HOST}:{PORT} --> Success
Send event TEST --> Incomplete
=> Trigger: TEST1 --> Launch number: XXXXXXX
=> 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 "0".