IBM ECM System Monitor 5.5.5.x – Integration in Service Management Tool

Integrating ESM in ServiceNow as an example using the WebHookExportIncident Task

April 15th, 2021

CENIT AG
Wohland, Michael

Table of Contents

Introduction

Overview

This guide describes in detail, how ESM can be integrated in a Service Management Tool. Examplarily we use “ServiceNow” in this guide. For the integration an ESM task named WebHookExportIncident will be used.

Disclaimer

The content of this document is based on ESM in version 5.5.5.1. The descriptions and guidelines in this document are for informational purposes only. Up-to-dateness, content completeness, appropriateness and validity for all possible scenarios cannot be guaranteed. All information is provided on an as-is basis. The author is not liable for any errors or omissions in this document or any losses, injuries and damages arising from its use.

If you are planning to setup or configure ESM or to adjust an existing installation, it is absolutely necessary to take into account current security whitepapers, release notes and announcements from the official IBM ECM System Monitor product documentation website.

Purpose of this guide

Most of the ESM users / customers want to integrate the ESM in their Service Management tool. ESM offers several possibilities for that. The WebHookExportIncident task offers a comfortable way to achieve that.

Requirements

- The Service Management tool must offer a REST API that allows JSON object to be posted. For example “ServiceNow” offers this option as a Web Service API.

- The ESM Server must be able to access the URL to the REST API.

Step by Step setup

1. Find the description of the REST API and create the message template accordingly

Check for the description of the API on the vendor pages. E.g. this is the URL of the Web Service API from “ServiceNow”: https://docs.servicenow.com/bundle/quebec-it-operations-management/page/product/event-management/task/send-events-via-web-service.html

On the “ServiceNow” page you can find this example for the JSON object with one record at a time:

```
{       "records":
[
     {
     "source":"SCOM",
     "event_class":"SCOM 2007 on scom.server.com",
     "resource":"C:",
     "node":"name.of.node.com",
     "metric_name":"Percentage Logical Disk Free Space",
     "type":"Disk space",
     "severity":"4",
     "description":"The disk C: on computer V-W2K8-dfg.dfg.com is running
out of disk space. The value that exceeded the threshold is 41% free
space.",
     "additional_info":{
                                                 'scom-severity':'Medium',
                                                 'metric-value':'41',

        'os_type':'Windows.Server.2008'
                                          }
      }
   ]
}
```

The default template in the WebHookExportIncident task looks like this:

$Timestamp: $Severity: $Value | $Message | {"text" : "$Timestamp: $Severity: $Value | $Message | $Error "} +

Incident internal fields are used with the following notation: $EntityType.FieldInCamelCase. They can be seen when double clicking on an incident. The field must be defined in CamelCase, meaning each word starts with a capital letter. E.g. SituationCfgId. Some of the fields e.g. Incident.ID might not be available at task execution since the entry is created when added to the DB.

The following EntityTypes are available:

- Incident (default - used if nothing but FieldInCamelCase is specified)
- Situation
- Sample
- ProbeConfig
- Agent
- Subsystem

The information must be combined with the needed JSON object to the actual template. This is just an example and may not fit in your environment. Some JSON object variables have been removed as they are not needed:

```
{       "records":
[
      {
      "source":"$Source",
      "event_class":"$Classification",
      "node":"$Agent.HostName",
      "metric_name":"$Probe.Name",
      "severity":"4",
      "description":"$Message",
      "additional_info":{ "This alert pertains to XYZ System - $Timestamp | $Error"
                                                }
       }
   ]
}
```

The template is completed, make sure to put it in a one line format after you have created it. We recommend to do that in an editor like notepad++ or vim.

```
{"records": [ { "source":"$Source", "event_class":"$Classification", "node":"$Agent.HostName", "metric_name":"$Probe.Name", "severity":"4", "description":"$Message", "additional_info":{ "This alert pertains to XYZ System - $Timestamp | $Error" } } ] }
```

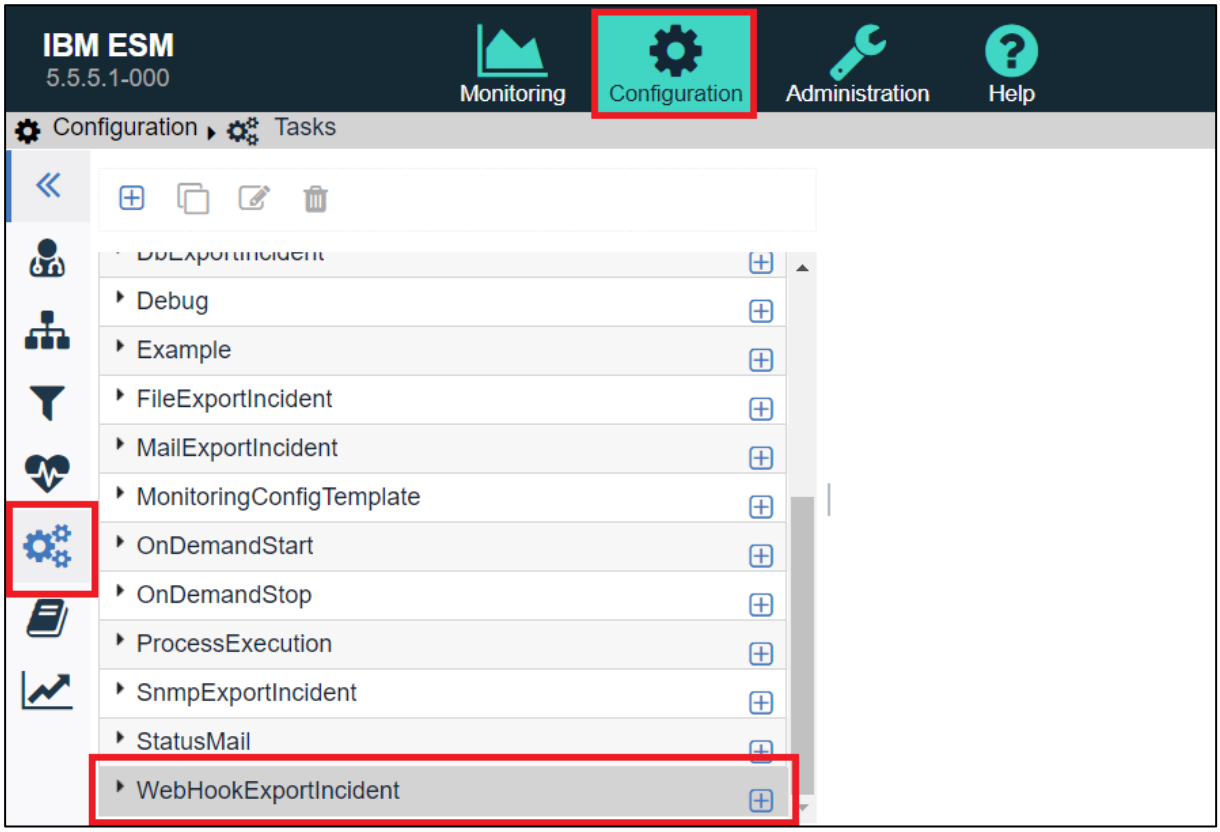
The severity could not be replaced in this case by using the $Severity info from the incident, as the information would be something like HARMLESS or CRITICAL. The object needs a number in this case. So depending on how many different severities for objects are needed in the Service Management Tool, you might end up using several ESM tasks (one for each severity).

2. Setup of the actual task

Once the template is created, the URL where the JSON object will be sent to, is needed. In case of “ServiceNow” an example is given on the above specified web page again. As you can see there are various URLs for single or multiple records that are provided in the JSON object. Because the task will be triggered by one incident only, the JSON object will only contain one record. Therefore the URL is specified with this default look like on the web page:

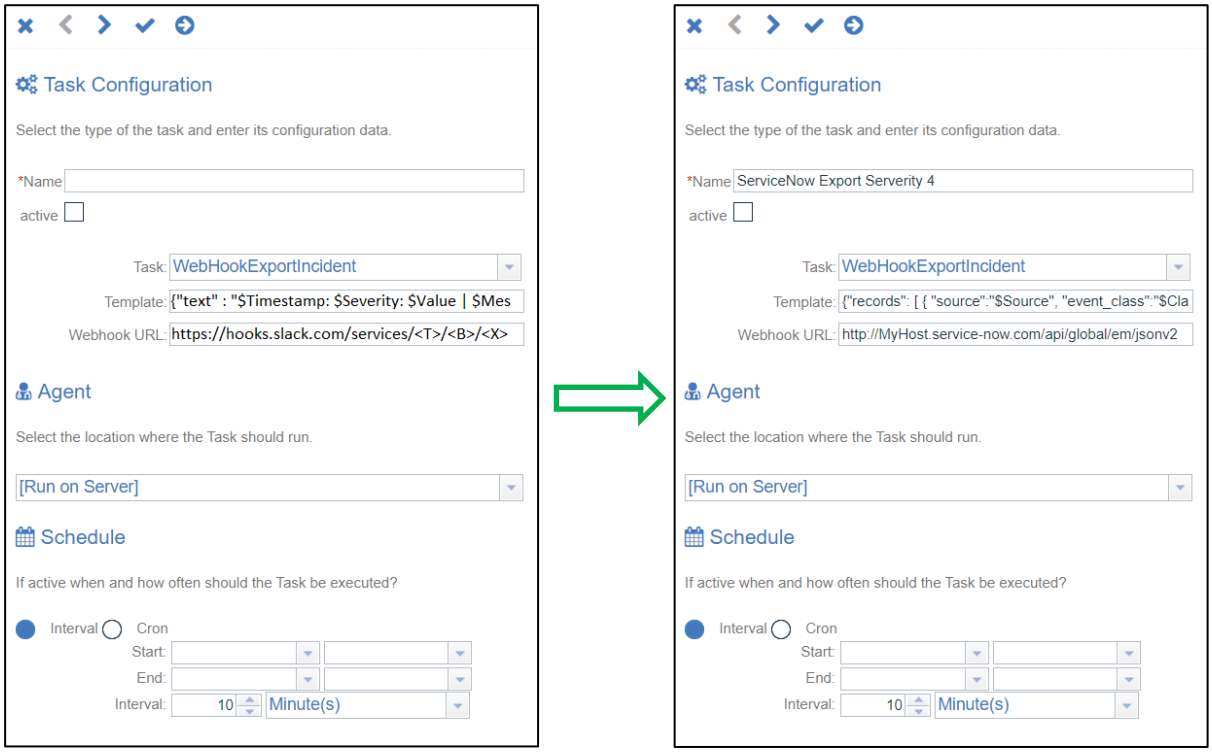
http://<instancename>.service-now.com/api/global/em/jsonv2

The actual task setup is done in the ESM console. Browse to the configuration and on the left to the tasks. There you will find the WebHookExportIncident task.



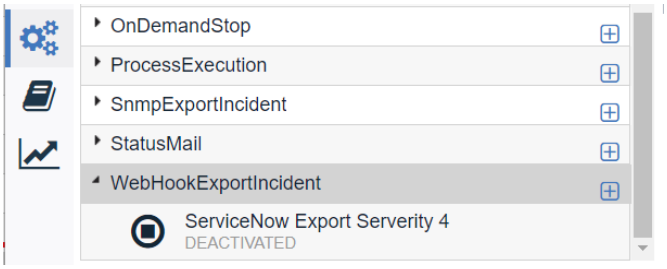
Click on the + on the right of the WebHookExportIncident task to open the task editor.

The editor looks like this:



- Specify a name for the task, e.g. ServiceNow Export Severity 4.
- Do not check the active button as the task will be triggered by an incident.
- Copy the template that has been created in the template section.
- Copy the matching URL in the Webhook URL.
- As Agent use [Run On Server] → The task will be executed on the actual ESM Server which is what we want.
- Because the task is triggered, the schedule part will not be used at all.
- Save the task by clicking on the hook button above the “Task Configuration”.

You should now see the saved task specified as “DEACTIVATED” below the WebHookExportIncident entry in the task list.

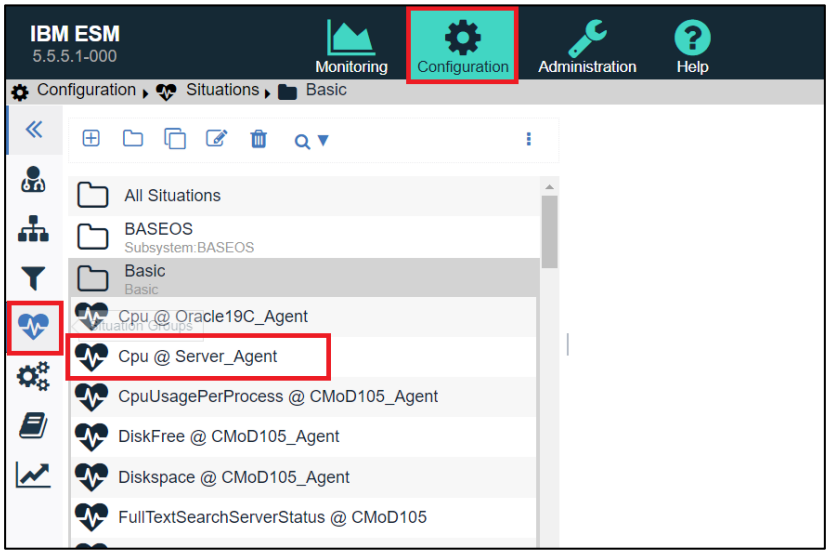


Repeat this step for tasks that use other severities, e.g. three more tasks for severities 1-3.

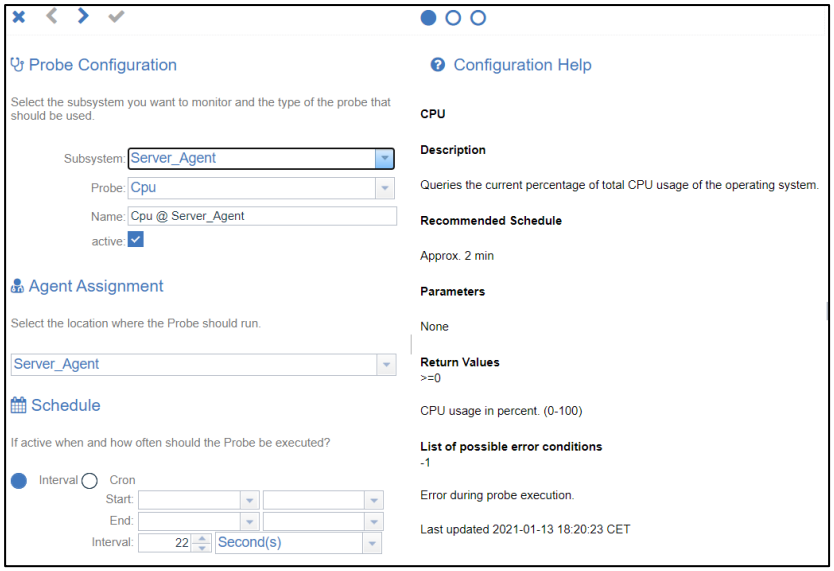
3. Trigger the task (forwarding to ServiceNow) once the incident is created

For forwarding the incident to the Service Management tool a task trigger must be defined. This is done in the situation editor. For each incident to be forwarded this must be done in the corresponding situation setup. As a simple example this is done with the "CPU @ Server_Agent" situation here.

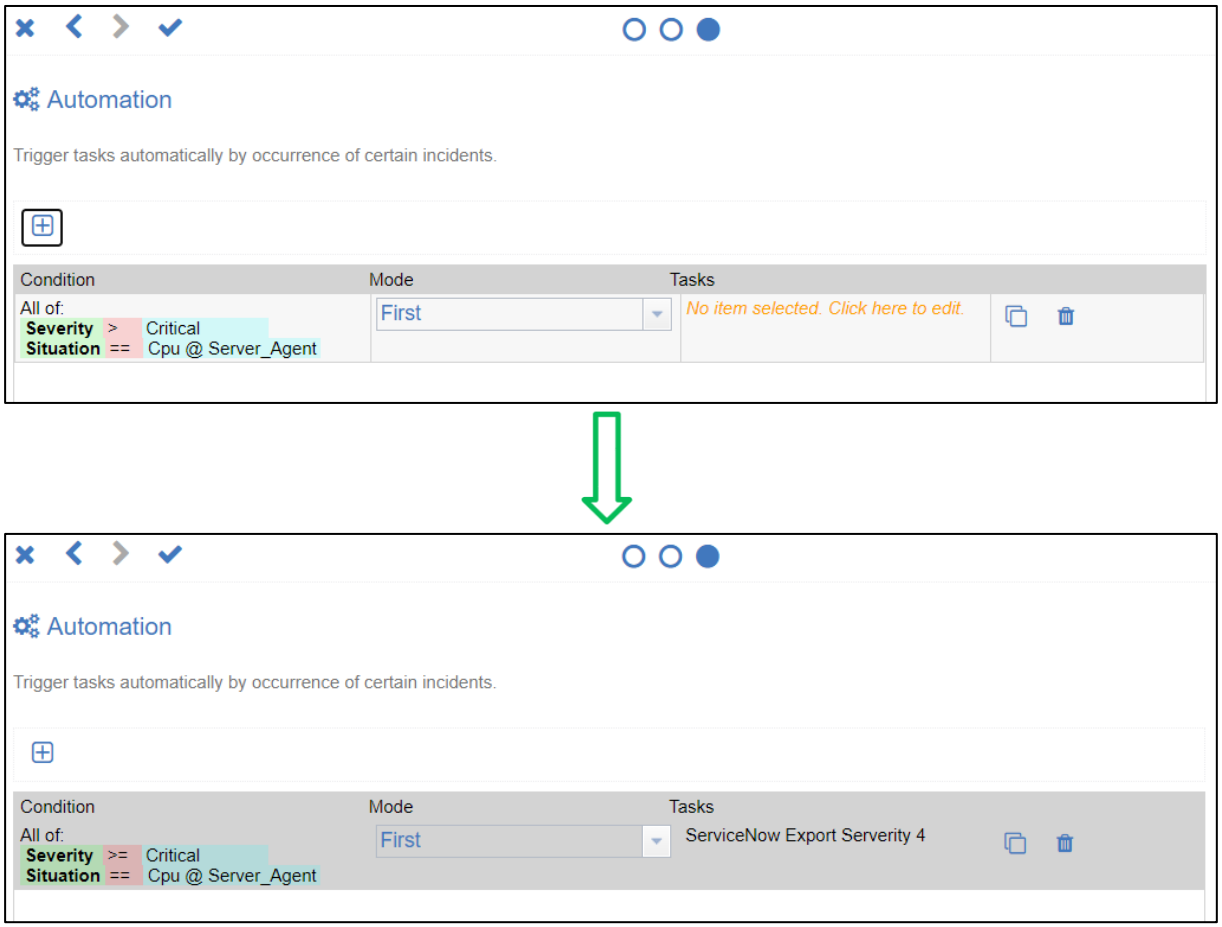
Select the configuration dashboard and switch to the "Situation Groups" on the left. Open the situation editor by double clicking on the situation for which you want to add the trigger.



The editor will open and look like this:



Switch to page 3 (Automation) of the editor. Per default no entry is given there. Click on the + to add a new setup.



Condition:

You should adjust the condition to your needs, but you should keep the entry “Situation == XYZ” at this point. Otherwise the condition will match all situations and the task is trigged from each incident that matches the condition.

Mode:

Select from the drop down between “First” or “Always”.

- First: will trigger the task only once – once the condition changes to a first match – so if the incident is currently having this condition already, nothing will be triggered.
- Always: Each time the condition matches the currently created incident, the task will be triggered.

Tasks:

Select the task you want to trigger once the condition matches from the drop down.

Save the situation setup by clicking on the hook button above the “Automation”. Repeat this for all incidents that must be forwarded. The integration is now complete.

Contact Information

If you have any questions please contact us at ECM.SystemMonitor@cenit.com.

CENIT AG
Phone: +49 711 7825 30
Email: ECM.SystemMonitor@cenit.com