|  |  |
| --- | --- |
|  |  |

# Probe – Selfmonitoring 1.5

**CA Services**

### Version 1.5

|  |
| --- |
| * Add alarm clear support. |

### Version 1.4

|  |
| --- |
| * Add alarm enrichment. * Add a supp\_key for every type of alarm. * Fix the subsystem\_id. |

## Introduction

Selfmonitoring has been created to do hub probe(s) monitoring. It has been created to work on a hub. Remote checking are only for UMP robot.

**The probe work with PERLUIM framework version 4.x**

## Features

* NisBridge monitoring (with HA support).
* Hub probe(s) monitoring (with HA support).
* Hub robot(s) spooler and intermediate state monitoring.
* Remote UMP monitoring.
* Distsrv jobs monitoring.
* Hub uptime monitoring.

When a problem is detected the probe trigger a new alarm. The probe also support alarm clear when the problem is solved (**only when it’s possible**).

Installation and configuration guide

### Setup section :

| **Key** | **Values** | **Description** |
| --- | --- | --- |
| **domain** | string | CA UIM Domain |
| **audit** | 1 - 0 | When audit is set to 1, the probe does not generate new alarms (cool to test in production the first time). |
| **callback\_retry\_count** | number | Number of retries of primary callbacks (getrobots and probeslist). |
| **output\_directory** | string | the name of output directory. |
| **output\_cache\_time** | number | the cache time in second for output directory. |

### Configuration section :

| **Key** | **Values** | **Description** |
| --- | --- | --- |
| **check\_nisbridge** | yes - no | Check Nis\_bridge state (Support HA) |
| **priority\_on\_ha** | yes - no | HA To rewrite 'alarm\_on\_probe\_deactivated' to 1 on every probe (if ha\_superiority is set to 'yes' on the probe configuration) |
| **alarms/intermediate** | 1 - 0 | Launch alarms when we detect intermediate robot. |
| **alarms/spooler** | 1 - 0 | Launch alarm when callback get\_info fail on one robot spooler. |
| **check\_hubuptime** | yes - no | Check for hub restart and generate a new alarm |
| **uptime\_seconds** | number | Uptime in second for check\_hubuptime |

UMP Robot (Wasp) monitoring

Setup ump monitoring on your primary and secondary hub. Just setup all UMP names in the “servers” key.

<ump\_monitoring>

servers = ump1robotname,ump2robotname

alarm\_callback = ump\_failcallback

alarm\_probelist = ump\_probelist\_fail

</ump\_monitoring>

**Note:** This section is optional

Reconfigure alarm as wanted in the “alarm\_messages” section of the probe configuration.

Distsrv monitoring

Monitoring of Distsrv deployment(s) and job(s).

<deployment\_monitoring>

job\_time\_threshold = 600

max\_jobs = 5000

</deployment\_monitoring>

**Note:** This section is optional

**Distsrv monitoring is an “experimental” feature (No clear support).**

Hub probe(s) monitoring

Setup your probes here. Callback is an optional key (no callback is the equivalent of probe down/up checkup). Set **alarm\_on\_probe\_deactivated** to 0 if you don’t want to trigger an alarm when the probe is offline.

**Warning:**  alarm\_on\_probe\_deactivated is rewrited to 0 by HA if you have priority\_on\_ha set to yes (configuration/priority\_on\_ha).

Set ha\_superiority to no if you don’t want HA failover to rewrite the profile configuration.

<probes\_monitoring>

<discovery\_server>

callback = get\_device\_statistics

alarm\_on\_probe\_deactivated = 0

ha\_superiority = yes

</discovery\_server>

<alarm\_enrichment>

callback = getStatistics

alarm\_on\_probe\_deactivated = 1

ha\_superiority = yes

</alarm\_enrichment>

<prop\_processor>

alarm\_on\_probe\_deactivated = 1

ha\_superiority = yes

</prop\_processor>

<nas>

callback = get\_info

alarm\_on\_probe\_deactivated = 1

ha\_superiority = yes

<check\_keys>

<pub\_subscribers>

<0>

name = NiS-Bridge

queue\_len = <<100000

</0>

</pub\_subcribers>

</check\_keys>

check\_alarm\_name = checkconfig\_nisbridge

</nas>

</probes\_monitoring>

Alarms configuration

Alarms message(s) are configurable in the alarms\_messages section. Variable(s) are setted in the Script (so refer to the below table to use variables for each of them).

### Alarms default fields name:

Origin, domain, source, dev\_id, usertag2, usertag1, supp\_key, probe, robot, rc

### Alarms custom field(s):

| **Callback** | **Variables** |
| --- | --- |
| **callback\_fail** | $callback, $probeName, $hubname |
| **probe\_offline** | $probeName, $hubname |
| **spooler\_fail** | $hubname |
| **intermediate\_robot** | $hubname |
| **nisbridge** | $hubname, $robotname, $nis, $ha |
| **distsrv\_deployment** | $jobid, $pkgName, $started, $diff, $hubname, $robotName |
| **distsrv\_maxjobs** | $max, $count, $hubname, $robotName |
| **ump\_probelist\_fail** | $robotname, $umpName |
| **ump\_failcallback** | $robotname, $umpName |
| **hub\_restart** | $second, $hubName |

### Create your own variable(s):

Add your own variable(s) in the code, just search for the callback name. When it’s done just use “**$customVar**” in the configuration file to get “**toto**” in the alarm message.

my $probe\_offline = $alarm\_manager->get('probe\_offline');

my ($RC\_ALARM,$AlarmID) = $probe\_offline->call({

probe => "$probe->{name}",

hubname => "$hub->{name}",

customVar => "toto"

});