

Process by name

Process by name probe returns information about process with the specified name and/or arguments.

- [Example](#)
- [Result](#)
- [Terminal counterpart](#)
- [Known problems and errors](#)

Example

The screenshot shows a configuration window titled "Monitoring". It contains several fields for setting up a probe:

- Agent:** Saymon Agent (ID:209) /Top Level/SAYMON Group/Hosting Group/inode saas/Saymon Agent
- Probe type:** Process by name
- Process:** nodejs
- Arguments contain:** Rest
- Period:** Probe period (input field) and seconds (dropdown menu)

| Field | Description |
|--------------------|--|
| Process* | Process name to find. The value has to match strictly to the process name to be found, e.g. for "nodejs" value - "process name matches <i>nodejs</i> ". |
| Arguments contain* | Find processes with arguments, that contain specified value. E.g. for "Rest" value - "one of the arguments of the process contains <i>Rest</i> ". |
| Period | Describes how often to perform the check (60 seconds by default). |
| * | To setup this probe one of the field is enough. If the both fields are filled the logical operator "AND" is applied, e.g.: "process name matches <i>nodejs</i> AND one of the arguments of the found processes contains <i>Rest</i> ". |

Result

After the check has been done the result is displayed in the data table:

| Data | | | | | | | | | | | | | | Table for field: -- Not selected -- | | Exact values <input type="radio"/> NO <input checked="" type="radio"/> | |
|--------|-----------|----------|-------|-----|------|-----------------|---|-----------------|--------------|---------------|-------------|----------------------|----------------------|-------------------------------------|--|--|--|
| name | processId | parentId | state | tty | nice | arguments.0 | arguments.1 | arguments.2 | bytesVirtual | bytesResident | bytesShared | lastTime | startTime | userCp | | | |
| nodejs | 29050 | 29022 | S | 0 | 0 | /usr/bin/nodejs | /usr/local/saymon/server/factors/lib/forked-actor-worker.js | RestServerActor | 1.3 G | 191 M | 9.22 M | 07.12.2017, 11:07:03 | 30.11.2017, 16:04:00 | 20.3 M | | | |

Rows: 1

| Field | Description |
|---------------|---|
| name | Name of the found process. |
| processId | Process ID number (PID). |
| parentId | ID number of the processs parent process (PPID). |
| state | <p>Process state code:</p> <ul style="list-style-type: none"> • D uninterruptible sleep (usually IO) • R running or runnable (on run queue) • S interruptible sleep (waiting for an event to complete) • T stopped, either by a job control signal or because it is being traced • W paging (not valid since the 2.6.xx kernel) • X dead (should never be seen) - the process is about to be destroyed • Z defunct ("zombie") process, terminated but not reaped by its parent • < high-priority (not nice to other users) • N low-priority (nice to other users) • L has pages locked into memory (for real-time and custom IO) • s is a session leader • l is multi-threaded (using CLONE_THREAD, like NPTL pthreads do) • + is in the foreground process group |
| tty | Terminal associated with the process. |
| nice | Nice value. This ranges from 19 (nicest) to -20 (not nice to others). |
| arguments.xxx | Arguments of the process, if any. |
| bytesVirtual | Total process virtual memory (virtual memory usage). |
| bytesResident | Total process resident memory. |
| bytesShared | Total process shared memory. |
| lastTime | Time when the process ran on a cpu and did something. |
| startTime | Time when the process started. |
| userCpu | Process cpu user time. |

| | |
|------------|--|
| systemCpu | Process cpu kernel time. |
| totalCpu | Process cpu time (sum of userCpu and systemCpu). |
| percentCpu | How much of the CPU the process is using (or "process cpu usage"). |

Raws is an additional field in which the array length is indicated.

Total number of rows is displayed when the array value is the result of the probe or the input data.

Terminal counterpart

This check is the counterpart of ps terminal command:

```
ps aux | head -1 ; sudo ps aux | grep 'nodejs.*Rest' | grep -v grep

USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
saymon   29050  3.7  4.7 1359092 194040 ?        S    Nov30 363:39 /usr/bin/nodejs /usr/local/saymon/server/actors/lib/forked-actor-worker.js
RestServerActor
```

Known problems and errors

At the time of writing this article, no such errors were found.