Table of Contents

Picom mechanism

What is a Picom ?

Picom stands for Process Intelligent COMponent. A picom is a component run as a task (PicomTask) that runs a user-defined script in the name of the user that is able to interact with Gateway and in particular to submit other tasks.

Initially, the picom was designed to be a very generic mechanism that would be specialized into different standard workflow control components (Loop, Switch/Case, Containers, Parsers). The mechanism has proven that it is much more powerful (business process integration, dynamic workflows, recursive workflows, AI oriented workflows...)

Principle

File-based dialog

The principle of the picom script is to do a file-based dialog with Gateway. The picom script reads in the runlog all the information :

- about itself,
- the context it's running in,
- the template tasks it can use as model for the new tasks it,
- the tasks it has spawned previously

The runlog is the center of the dialog between Gateway and the picom script.

The runlog contains :

- the picom script (python)
- the picom description (JSON)
- the tasks templates (JSONs)
- the log of the different script executions

```
$ tree -L 2 /home/michael/hpcgateway/runlog/michael/2018/02/21/picom_4857
/home/michael/hpcgateway/runlog/michael/2018/02/21/picom_4857
├── picom.json
├── picom.log
├── script.py
├── steps
│   ├── 0
│   ├── 1
│   ├── 2
│   ├── 3
│   └── 4
└── templates
```

```
    ├── template_0000.json
    ├── template_0001.json
    └── template_0002.json
```

There is also a sub-directory for each 'step' of the picom execution. Each step sub-directory contains:

- the context description (read and written by the script)
- the description of all the tasks before the script execution
- the description of the tasks the must be submitted after the script execution (written by the script)
- the status of the picom task (written by the script)

```
$ tree -L 2
/home/michael/hpcgateway/runlog/michael/2018/02/21/picom_4857/steps/3
/home/michael/hpcgateway/runlog/michael/2018/02/21/picom_4857/steps/3
├── context.json
├── lastTasks
│   ├── task_0000.json
│   ├── task_0001.json
│   ├── task_0002.json
│   ├── task_0003.json
│   └── task_0004.json
├── nextTasks
│   ├── task_000.json
│   ├── task_001.json
│   ├── task_002.json
│   ├── task_003.json
│   └── task_004.json
└── status.json
```

Context

The executions of the picom tasks and its sub-tasks is influenced by the *context* of the execution. The context is keep the trace of the executions and it propagated to the sub-tasks of the picom task.

Step and maxStep

The *step* correspond to the different executions of the picom script.

The step is limited by a field *maxStep* in the context (default is 100).

Depth and maxDepth

The *depth* is the depth of the task hierarchy. The picom task that is created by the user has a depth 0 and its immediate children have a depth 1.

The depth is limited by a field *maxDepth* in the context (default is 3).

The depth and its limitation are crucial in the recursive picoms to avoid the picoms to run wild.