How to configure a JobScheduler Master and Agent to work together

Scope

- This article explains configuration of the Classic JobScheduler Agent

  `FEATURE AVAILABILITY ENDING WITH RELEASE 1.10`
- For configuration of the JobScheduler Universal Agent see JobScheduler Universal Agent - Installation & Operation

  `FEATURE AVAILABILITY STARTING FROM RELEASE 1.10`

JobScheduler and Agent

- The architecture for JobScheduler and Agents forces some prerequisites.
- Make sure that the JobScheduler can communicate with the *Agent JobScheduler* and vice versa.

JobScheduler Agent configuration

For this, edit the `./config/scheduler.xml` of the *Main JobScheduler*:

```
...
<security ignore_unknown_hosts="yes">
    ...
    <allowed_host host="[host or IP of the Agent JobScheduler]" level="all"/>
</security>
...
```

and edit the `./config/scheduler.xml` of the JobScheduler Agent :

```
...
<security ignore_unknown_hosts="yes">
    ...
    <allowed_host host="[host or IP of the Main JobScheduler]" level="all"/>
</security>
...
```
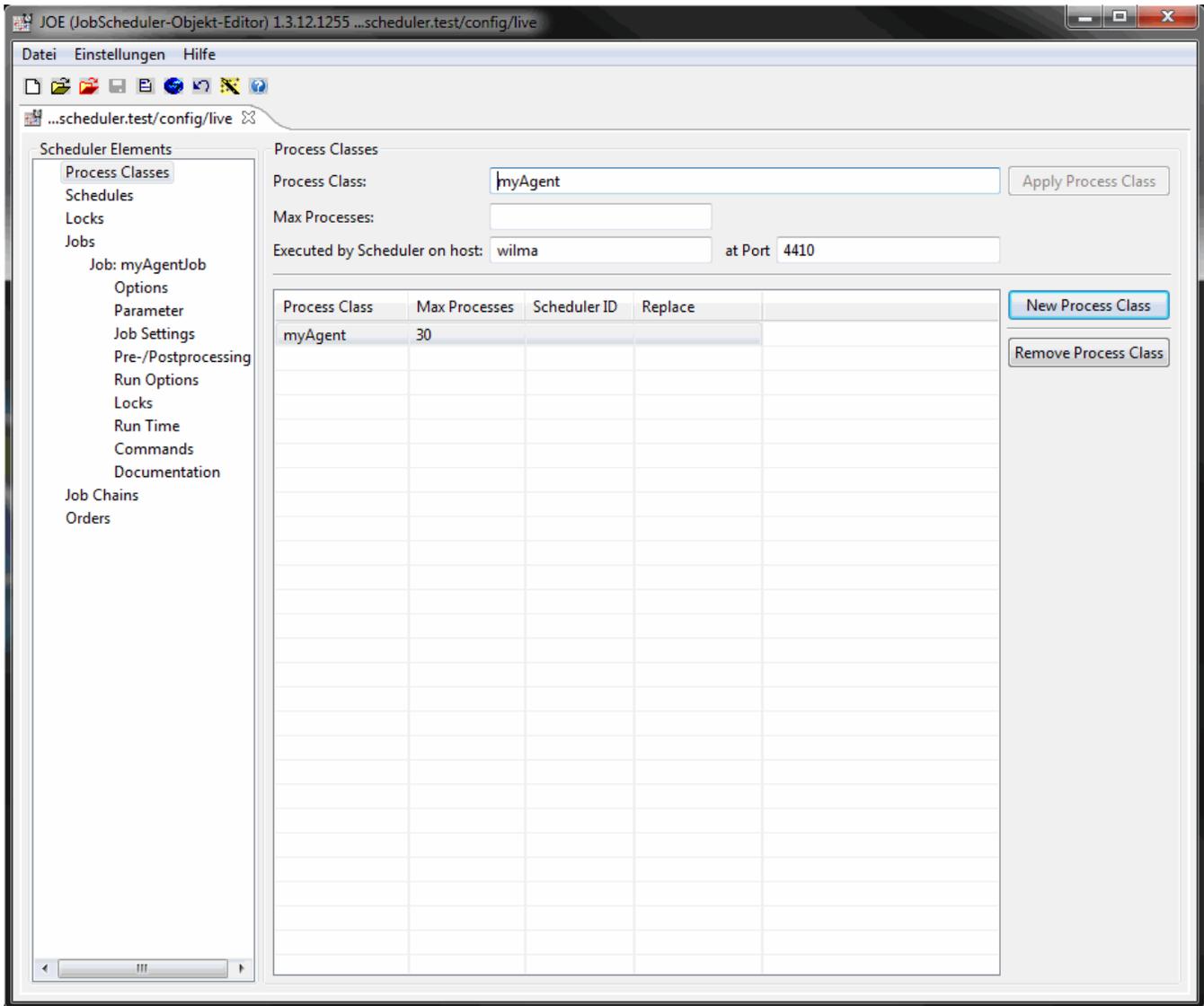
- Make sure that no firewall restricts the communication.
- See also http://www.sos-berlin.com/doc/en/scheduler.doc/remote_execution.xml
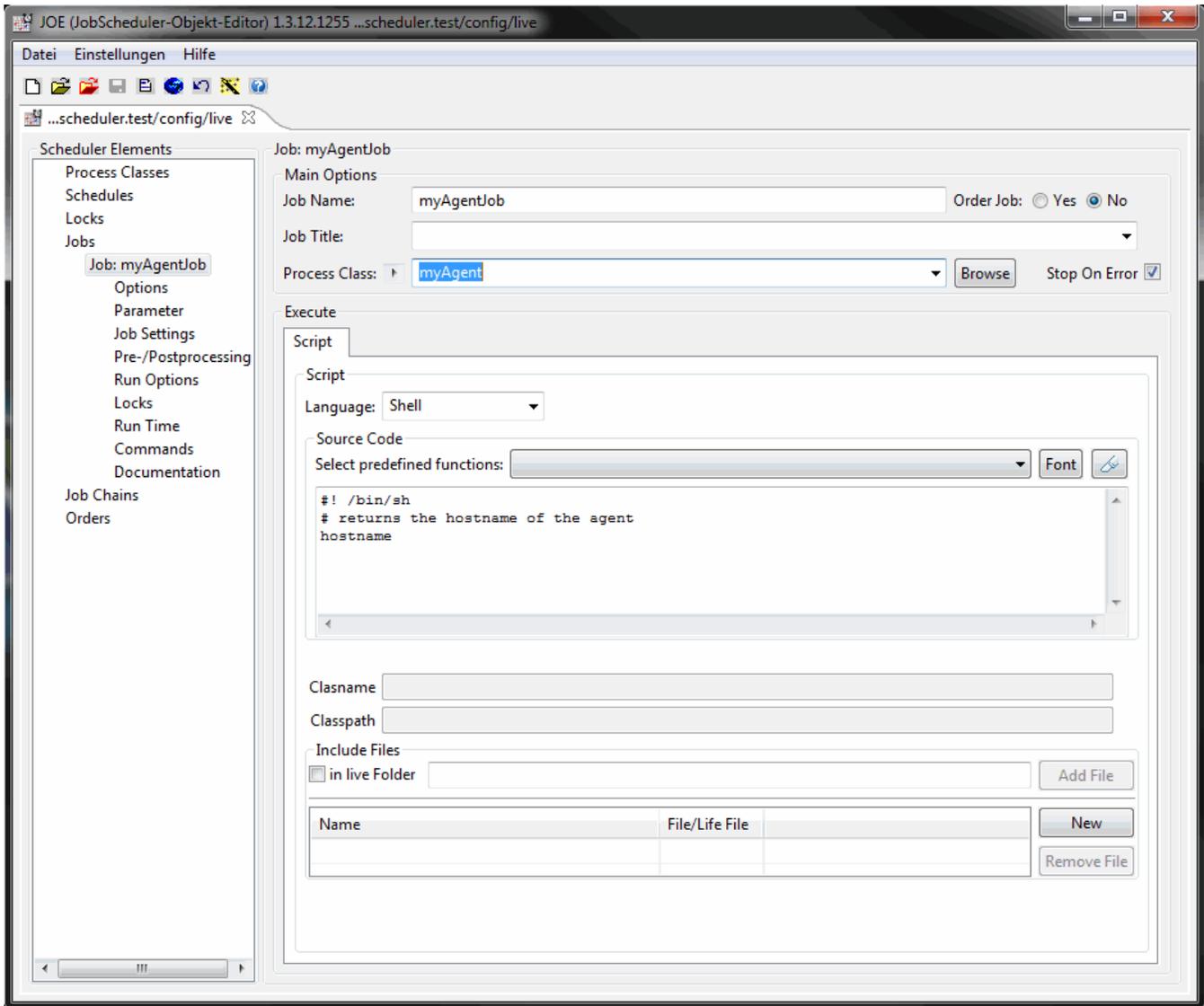
Process Class configuration

Now you need a *Process Class* and a Job in the Hot Folder (`./config/live`) of the JobScheduler to test the JobScheduler Agent.

In our example, let the JobScheduler Agent's hostname be `wilma` and its port `4410`:

- open *JOE - JobScheduler Object Editor* (JobScheduler Object Editor)
  Use the shortcut *Ctrl-D* to open the *Hot Folder* in *JOE*. You open this dialog from the menu *File->Open Hot Folder* or with the red folder icon.
  The Hot Folder should be already selected, otherwise please navigate to `./config/live`.
- Create the process class.
  Click on *Process Class* in the left tree of the JobScheduler elements.
  With *New Process Class* on the right hand side you create a process class which is stored with *Apply Process Class* after having entered the name `myAgent`, the JobScheduler Agent's host and port will be `wilma:4410`.
  A file `./config/live/myAgent.process_class.xml` is created with the following content:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<process_class max_processes="30" remote_scheduler="wilma:4410"/>
```

Job configuration

- Create the job that uses the above process class.
- Click on *Jobs* in the left tree of the JobScheduler elements.
- With *New Standalone Jobs* on the right hand side you create a new job in the list of the jobs and in the left tree. Select it to open an empty job configuration element.

Enter in the field *Job Name* e.g. `myAgentJob` and in the *Source Code* text area the lines:

```
#! /bin/sh
# returns the hostname of the agent
hostname
```

- In order to have this job executed on the JobScheduler Agent, the job should be assigned the above process class.
- Assign the field *Process Class* the above `myAgent` process class.
- You store this configuration with the shortcut *Crtl-S* or with the *File->Save* function.

A file `./config/live/myAgentJob.job.xml` is created with the content:

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<job process_class="myAgent">
    <script language="shell">
        <![CDATA[
#! /bin/sh
# returns the hostname of the agent
hostname
        ]]>
    </script>
    <run_time/>
</job>
```

- The value in the *process_class* attribute of the *job* element should be assigned a relative path or absolute path to an existing process class.
  - An absolute path starts at the ./config/live folder.
- In the above example
  - a relative path is used and the job finds the *process_class* in the same folder.
  - you could use the absolute path *<job process_clash1. "/myAgent">* as well.
- For example, if the job is stored in *./config/live/myProject/myAgentJob.job.xml* then you can use *<job process_class"/myAgent">* or *<job process_class="../myAgent">*.
- Start the job `myAgentJob` in the *How to configure JOC - an overview* and check the task's log. You should see a line like this:

```
2011-10-10 15:26:47.237 [info]   wilma
```