Job Configuration

The scheduler and its behaviour for the various job types is fully customizable using an XML file. This file controls the required permissions for the various actions available using the `Scheduled jobs` tool. But is also includes the queues definitions as used by the scheduler.

State machine in detail

A more detailed view of the Scheduler state machine is needed here than the one outlined in Distributed scheduler



Tool-setup XML file

The Approval mechanism is part of the scheduler. All jobs requiring approval will be placed in a new state 'Pending' until it is approved before the state is advanced to 'Scheduled'.

To support approvals require all job-tools, the scheduler and the job-management tool work from a common set of tool-configurations. The job-tools need configuration settings on their approval requirements that will normally vary per user-level of the user submitting the jobs. The configuration settings include variables to define what user-levels may do the approvals, how many jobs may be submitted and the number of jobs that may be submitted without an approval.

Since these configuration settings differ per user-level and per job-type, the number of variables involved and the fact that both the job-tools and the scheduler require access to them strongly favour a centralized job-configuration file. Such an approach would also allow a customer to make (revision persistent) changes to meet his approval requirements.

The file defined for this purpose is `/opt/yce/etc/tool_setup.xml`. It will initially be maintained manually, an editing tool could be developed at a later date. To prevent overwriting this file during software updates, the `~/etc/tool_setup.xml` will not be distributed. Instead a 'master' file with default settings is supplied at `/opt/yce/system/tool_setup.xml` which will be copied to the `~/etc/` directory should it be missing.

Closely linked to the approval mechanism is the introduction of a job change-id. Its use and validation is very similar to the approval configuration variables and will therefore be included in the xml file.

The user-levels in YCE are incremental in authorization and are defined by the user-group the user is a member of. Distinction must be made between the users global user-level and the client-type specific user-level. In a user-group, the user may be assigned an 'engineer' user-level (3) globally, but access to the nodes belonging to a specific client_type (architecture / network) could be reduced to 'operator' (2). When submitting jobs for nodes it is the effective access level that defines the approval settings, not his global user-level.

Access authorization level definition (user-level and client-level)

User-level	name	remark
0	disabled	cannot login

User-level	name	remark
1	browser	cannot submit jobs
2	operator	
3	engineer	
4	modeler	aka designer
5	manager	
6	system	same as manager

The section below shows the tool-configuration entries for one of the job-tools, the 'command_job'.

```
<tool_setup>

    <command_job job_type="command_job" queue="yce">
        <auditors levels="3456" />
        <notify pending="yes" cancel="yes" suspend="yes" />
        <approvals name="2" levels="3456" threshold="0" limit="10" />
        <approvals name="3" levels="3456" threshold="0" limit="50" />
        <approvals name="4" levels="3456" threshold="2" limit="100" />
        <approvals name="5" levels="3456" threshold="2" limit="100" />
        <approvals name="6" levels="3456" threshold="0" limit="8" />
        <change_id option="2" hint="C000xxxxxx">
            <validation>^C000(\d){6}$</validation>
            <validation>^O000(\d){6}$</validation>
            <validation>^O000(\d){6}\-(\d){3}$</validation>
            <validation>^T000(\d){6}$</validation>
        </change_id>
    </command_job>

</tool_setup>
```

The configuration file has a 'tool_setup' section per job-type. Each of the job-generating tools in NetYCE has a job-type for which its configuration can be defined in its corresponding section. The example above shows a possible configuration for the 'command_job' tool.

The line (or 'record') defining 'auditors' has an attribute, 'levels', that defines who can manipulate the scheduled job. The 'levels' value forms a list of numeric privilege levels that are allowed to change the state of a job: set it for 'now', 'cancel', 'suspend' or 'resume'. It controls the “Select” checkbox in the Job-list tool. It does not control the approvals. When the 'level' value has the numbers '3', '4', '5' and '6' (producing “3456”), the checkbox is available for 'engineer' through 'system' level privileges. The privilege level of a user can depend on the client-type of the node for which the job was submitted.

The 'name' in the approvals record refers to a 'user-level', 1-6, corresponding to an 'operator' through 'system' privilege levels. However, in this case it is the permission level the user-group has for nodes of a client-type. This level is also referred to as the effective client-level.

The 'levels' attribute of the 'approvals' record is a list of user/client-levels that can 'approve' a scheduled job. When the operators permission level is not listed, no approval checkbox wil be available in the “Approval” column of the Job-list tool. Jobs submitted by the same user cannot be approved.

In the same 'approvals' record, the 'limit' controls the total number of jobs in the scheduler of the job-type that are in a 'scheduled' or 'pending' state. When submitting new jobs to the scheduler, they can either be refused should no approval levels be defined ('levels' attribute is empty) or accepted and set 'pending' awaiting approval.

Likewise, the 'threshold' defines how many of these jobs may be submitted to the scheduler before new jobs are set 'pending' requiring an approval.

Default job-type

The above example shows the full-definitions for just one tool, the full file will contain definition sections for many of these tools. These sections can be simplified using a <defaults/> section. Its usage is demonstrated below. Any value or section omitted in the tool-definition is taken from the <default> section.

```
<tool_setup>

    <defaults job_type="default" queue="yce">
        <auditors levels="3456" />
        <notify pending="yes" cancel="yes" suspend="yes" />
        <approvals name="1" levels="3456" threshold="0" limit="0" />
        <approvals name="2" levels="3456" threshold="0" limit="0" />
        <approvals name="3" levels="3456" threshold="0" limit="0" />
        <approvals name="4" levels="3456" threshold="0" limit="0" />
        <approvals name="5" levels="3456" threshold="0" limit="0" />
        <approvals name="6" levels="" threshold="0" limit="0" />
        <change_id option="2" hint="C000xxxxxx">
            <validation>^C000(\d){6}$</validation>
            <validation>^O000(\d){6}$</validation>
            <validation>^O000(\d){6}\-(\d){3}$</validation>
        </change_id>
    </defaults>

    <startup_config job_type="startup_config">
        <auditors levels="456" />
        <change_id hint="T000xxxxxx">
            <validation>^T000(\d){6}$</validation>
        </change_id>
    </startup_config>

<tool_setup>
```

When the <default> does not include a definition either, hard-coded defaults are used. The same applies when the tool_setup.xml file is missing or cannot be parsed.

The default settings for a missing job-tool entry are:

```
     <queue="yce" />
     <auditors levels="3456" />
     <approvals name="1" levels="3456" threshold="0" limit="0" />
```

```
    <approvals name="2" levels="3456" threshold="0" limit="0" />
    <approvals name="3" levels="456"  threshold="0" limit="0" />
    <approvals name="4" levels="56"   threshold="0" limit="0" />
    <approvals name="5" levels="56"   threshold="0" limit="0" />
    <approvals name="6" levels="56"   threshold="0" limit="0" />
    <change_id option="1" hint="optional">
```

Job configuration parameters

What do all these parameters control? Well, they allow you to tune the approval and auditor workflow and permissions to be tailored to your organization.

Element	Attribute	Values	Description and use
name			
	job_type	e.g. `command_job`	Name of the job, hardcoded in the job tool.
	queue	`yce`	Assigned queue name. Must exit in <queue> section
auditors			
	levels	e.g. 3456	The list of user-levels that may perform the actions (cancel, now, suspend, resume) on the submitted job. The user level refers to the level of access granted to the client-type of this job
notify			
	pending	`yes` or `no`	Specifies whether notification email must be sent tot the user-group address(es) informing of new jobs requiring approval
	cancel	`yes` or `no`	Specifies whether the job owner is notified with an email on the cancellation of his job. when the owner cancels the job, no notification is sent.
	suspend	`yes` or `no`	Specifies whether the job owner is notified with an email on the suspension of his job. Including when the job expires in the “Waiting” state. Should the owner suspend the job, no notification is sent.
change_id			
	option	`0`, 1, or 2	Indicates if the inclusion of the Change-id with the job is required. A value of `0` indicates the field will not be shown, a '1' shows the Change-id field but its use is optional, a value of 2 requires a value for the Change-id.
	hint	e.g. `C000xxxxxx`	Displays the value as a 'suggestion' for the Change-id when the field is blank. Useful for showing the format of the requested value.
	validation	e.g. `^C000(\d){6}$`	The value is a regular expression against which the entered change-id is tested. Non-matching Change-id's are rejected. Multiple validation values can be used to test the Change-id, they are all tried until a match is found. When a Change-id is entered, mandatory or not, the validation must succeed in order to accept the job.

> Need a table of available tool names (tool_types).

Queue definitions

The `tool_setup.xml` file also contains the definitions of the various scheduler queues. The default set is:

```
    <queues>
        <queue name="evpn" done_age="180" cancel_age="1800" job_int="20"
max_run="50" max_wait="3600" />
        <queue name="ios" done_age="180" cancel_age="1800" job_int="5"
max_run="20" max_wait="3600" />
        <queue name="yce" done_age="180" cancel_age="1800" job_int="2"
max_run="50" max_wait="3600" />
    </queues>
```

See the section on queue configuration in Queue Operation

From:
https://wiki.netyce.com/ - **Technical documentation**

Permanent link:
https://wiki.netyce.com/doku.php?id=guides:reference:jobs:job_configuration

Last update: **2024/07/03 12:31**