Event handling for job dependencies

Dependencies of job chains implemented by events

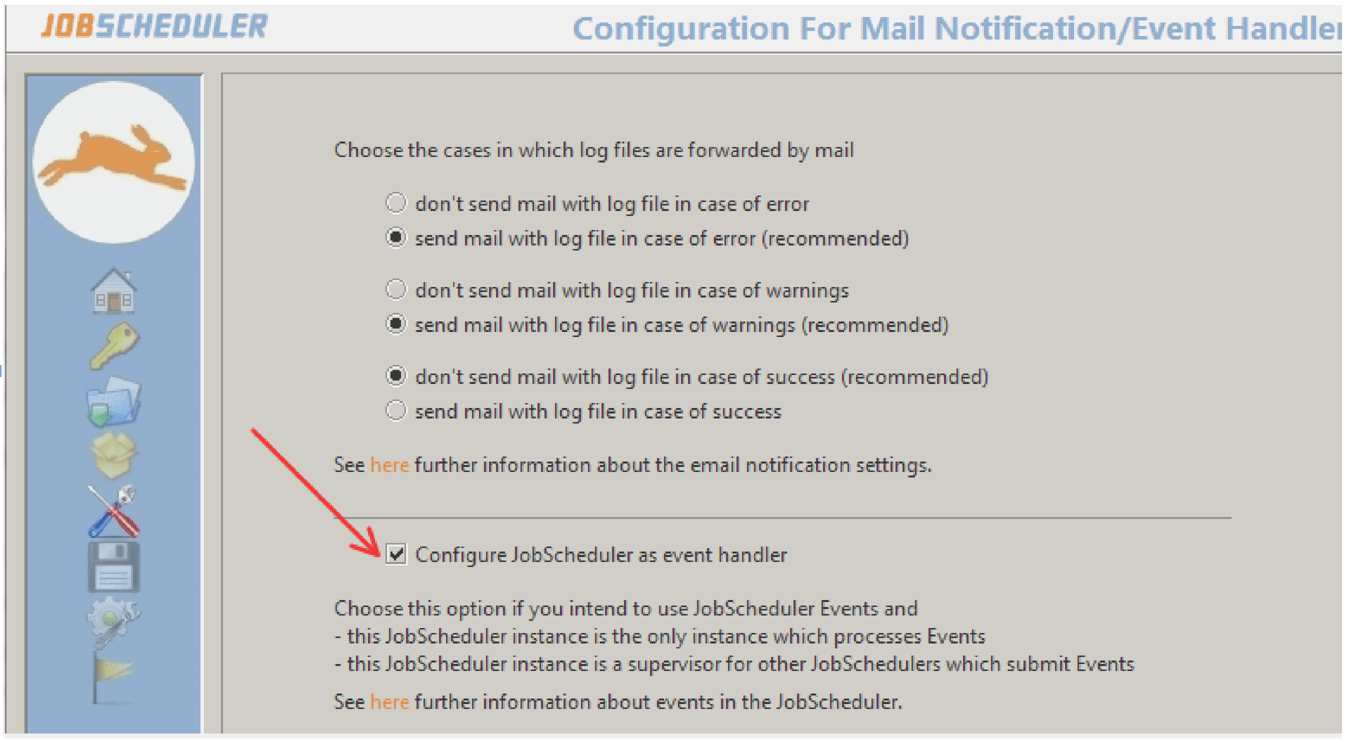
Event handlers are one of the methods available for implementing dependencies between JobScheduler job chains. Whenever an event is created an entry is made in the *scheduler_events* database table. The event handler contains the conditions that define whether an order is to be started for a job chain when an event is detected.

Events are processed by a JobScheduler Instance. This could be a Supervisor JobScheduler if a Workload JobScheduler is registered with that instance otherwise the Workload JobScheduler is used.

The events can be monitored with the JOC Cockpit web interface, which replaces the JID - JobScheduler Information Dashboard interface, which is deprecated as of JobScheduler Release 1.12.0.

Event Service

The *event service* has to be installed before events can be used. The *event service* is part of the JobScheduler installation if it has been configured during the installation procedure.



Choose the cases in which log files are forwarded by mail

- ○ don't send mail with log file in case of error
- ● send mail with log file in case of error (recommended)
- ○ don't send mail with log file in case of warnings
- ● send mail with log file in case of warnings (recommended)
- ● don't send mail with log file in case of success (recommended)
- ○ send mail with log file in case of success

See here further information about the email notification settings.

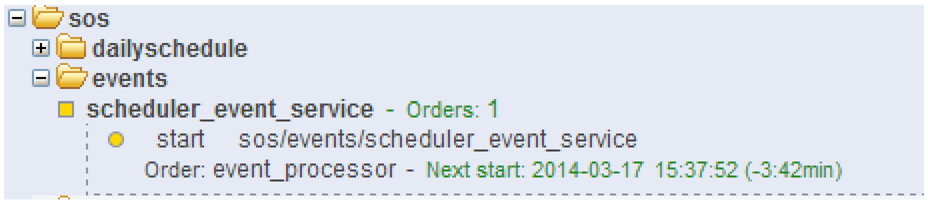
[✓] Configure JobScheduler as event handler

Choose this option if you intend to use JobScheduler Events and
- this JobScheduler instance is the only instance which processes Events
- this JobScheduler instance is a supervisor for other JobSchedulers which submit Events

See here further information about events in the JobScheduler.

The Event Service can also be installed in an existing JobScheduler instance. For more information see How to install the Event Service feature in JobScheduler.

The *scheduler_event_service* job chain is run at predetermined intervals or whenever an event is created. The event service checks all `.xml` data files in the `$SCHEDULER_DATA/config/events` folder for information matching the active events which are stored in the *scheduler_events* database table . All job chains within the actual live folder may be concerned.


sos
dailyschedule
events
scheduler_event_service - Orders: 1
start sos/events/scheduler_event_service
Order: event_processor - Next start: 2014-03-17 15:37:52 (-3:42min)

Event handler

The event handler is an `.xml` data file which contains the definitions of the events and the commands which are to be triggered .

For each job chain started by events we need an action to be defined in the event handler. An action has the two parts:

- Events ("if-clause")
- Command ("then-clause")

An event is uniquely defined by *event class* and *event ID*. Together with and/or/not the events of a group are combined to the condition.

In the command part the Order is defined to start the Job Chain as soon as the condition becomes `true`. It is also possible to use the *add event* command to create new events.

Every event which is used in the condition has to be deleted from the database with the *remove event* command .

Handling events in jobs / job chains

As part of the JobScheduler installation there are some Java classes to handle events:

- *JobSchedulerSubmitEventTaskAfterMonitor*
  - *JobSchedulerSubmitEventMonitor* can be configured as a monitor for (not shell-) jobs to submit an event.

- JobSchedulerSubmitEventJob
  - *JobSchedulerSubmitEventJob* can be configured to submit an event.
- JobSchedulerDequeueEventsJob
  - Dequeues Events. This job is triggered by orders or by standard job starts.
- JobSchedulerExistsEventJob
  - This job checks if certain event exists. The job processes orders which are configured with an event specification. Depending on whether these events exist or not the order will be put into the next_state or into the error_state.
- JobSchedulerCheckEvents
  - Check if events exist. This job is executed order driven.
- JobSchedulerEventJob
  - Process events. This job is executed order driven

Documentation

For more information about eventing see the reference guide JobScheduler Events, Definition and Processing