



Master-slave configuration for Scheduler

Revision 20200601

NOTE

This document is confidential and proprietary of **Denodo Technologies**.
No part of this document may be reproduced in any form by any means without prior written authorization of **Denodo Technologies**.

Copyright © 2022
Denodo Technologies Proprietary and Confidential

Goal

This document describes how to configure two instances of Scheduler (master/slave) and synchronize the data. Additionally we want to create a method to detect when the master server is down to use the slave as a temporary master server.

Content

In order to keep synchronized master and slave servers, there are two possible ways to do it:

1. Manually: Use the `import.sh` and `export.sh` scripts available under `<DENODO_HOME>\tools\scheduler`. Use the `export` script to export the scheduler's metadata from the master server and then use the `import` script to import the exported metadata into the slave server.
2. Automatically: Take advantage of Scheduler API to create a custom handler. In this custom element the connection information for both servers (host, port, user, password) is needed as input. Additional input parameters can be in place, for instance, to specify which projects/jobs must be synchronized. This custom element can be configured as part of a job to perform a periodic synchronization.

Important note: Take into account that when importing the metadata from the master server the different jobs will be enabled and both servers will have all jobs enabled. After the import process all jobs must be disabled on the slave server to avoid the simultaneous execution of the jobs. Even disabling all the jobs some jobs could start if they are triggered between the import process and disabling the jobs. To avoid this problem, the synchronization job must be launched when no job is scheduled during its execution.

As a template for the custom handler, follow these steps:

- Create a `schedulerManager` instance for the master server. (e.g `managerA`)
- Create a `schedulerManager` instance to the slave server. (e.g `managerB`)
- The code

```
managerB.importData(managerA.exportData(false, true, true, false), true);
```

will create an export of the metadata from the master instance and will export it to the slave instance.

- After the import the Scheduler API can be used to disable all the jobs in the slave instance.

Once both servers are synchronized, we want to always keep one of the servers available. A ping job can be executed periodically in the slave server to check if the master server is alive and, if not, enable all the jobs in the slave server to keep running the scheduled jobs.

To do this, a custom handler can be created to execute a dummy operation (e.g. obtain the configuration of any element) on the master server and to enable the slave jobs if there is an error with the operation on the master server.

