

Grouper Daemon

Wiki Home	Grouper Release Announcements	Grouper Guides	Grouper Deployment Guide	Community Contributions	Internal Developer Resources
---------------------------	---	--------------------------------	--	---	--

- [Bad Membership Finder](#)
- [Daemon configuration](#)
- [Grouper automatically fix attribute corruption](#)
- [Grouper daemon "other job"](#)
- [Grouper Daemon - job history chart](#)
- [Grouper GSH change log consumer](#)
- [Syncing groups between group management systems](#)
- [Universal Subject Daemon Utility \(USDU\)](#)

The Grouper Daemon is a process that can handle many tasks. In Grouper v2.4+, all daemon jobs are visible in the Grouper UI.

- Grouper daemon has a component called [Grouper Loader](#) that can automatically provision Grouper memberships from external SQL sources
- Grouper daemon is a container that runs in tomcat
- This daemon is required for all deployments, even if you are not using it to provision Grouper memberships from external SQL sources
- There are a lot of daemons in grouper including
 - [Disable expired memberships or to enable memberships](#) which are enabled in the future
 - Delete old audit and notification logs (configured in grouper-loader.properties)
 - Massage the notification logs so they have a sequential index number
 - Validate [Grouper Rules](#) and mark invalid ones as invalid. (v2.0)
- Notification consumers (callbacks) can be registered as a daemon. Grouper will keep track of which [change log](#) number they have successfully processed so the daemons can maintain state across Grouper Loader restarts
- The Grouper Loader keeps database logs in the grouper_loader_log table. These are periodically cleaned out based on configuration
- There is a [daily report](#) that can be emailed out to Grouper admin which details the state of the registry and the status of all daemon jobs from the last day
- The [PSP changelog provisioning](#) can be enabled as a daemon process (v2.1)
- In v2.3+, the daemon is pre-configured to use a database to store job schedules instead of storing them locally in memory.
 - By default, the Grouper database is used. These are the grouper_QZ_* tables.
 - You can run the daemon on multiple machines and jobs will be spread among them all automatically. Be sure to keep your loader configuration (grouper-loader.properties) the same on all machines.
 - You can also use the daemon to schedule any custom jobs that you may have. To do so, add the following configuration in grouper-loader.properties:

```
#####
## Other jobs
##
## Configure other jobs.
## "jobName" is the name of your job.
## Class must implement org.quartz.Job.
## Priority is optional
#####

# otherJob.jobName.class =
# otherJob.jobName.quartzCron =
# otherJob.jobName.priority =
```

- In Grouper v2.4+, all daemon jobs are visible in the Grouper UI. The screen below is linked from the "Miscellaneous" link in the Quick Links section. You can also see their associated logs from the grouper_loader_log table by clicking on a job name.

Home > Miscellaneous > All daemon jobs

All daemon jobs

Filter for:
☐ Show extended results?

Next refresh: Maximum refreshes reached

Job name	State	Last run status	Actions
CHANGE_LOG_changeLogTempToChangeLog	ENABLED	SUCCESS	<input type="button" value="Job actions"/>
CHANGE_LOG_consumer_grouperRules	ENABLED	SUCCESS	<input type="button" value="Job actions"/>
CHANGE_LOG_consumer_syncGroups	ENABLED	SUCCESS	<input type="button" value="Job actions"/>
LDAP_GROUPS_FROM_ATTRIBUTES__attributesStem:myLdapGroupFromAttributes__6d355f531f0441b9b522ca2da9263470	ENABLED	SUCCESS	<input type="button" value="Job actions"/>
LDAP_GROUP_LIST__testall:groups:ldap group list__8b9b945fee7940c3aeb82027ddffce3f	ENABLED	SUCCESS	<input type="button" value="Job actions"/>
LDAP_SIMPLE__testall:ldap simple__8ee32a7a0ffb4b72bfc558545f25d526	ENABLED	SUCCESS	<input type="button" value="Job actions"/>
MAINTENANCE__builtinMessagingDaemon	ENABLED	SUCCESS	<input type="button" value="Job actions"/>
MAINTENANCE__enabledDisabled	ENABLED	SUCCESS	<input type="button" value="Job actions"/>
MAINTENANCE__grouperReport	ENABLED	ERROR	<input type="button" value="Job actions"/>
MAINTENANCE__rules	ENABLED	SUCCESS	<input type="button" value="Job actions"/>
MAINTENANCE_cleanLogs	ENABLED	SUCCESS	<input type="button" value="Job actions"/>
MESSAGE_LISTENER_messagingListener	RUNNING 1 minutes 2 seconds	SUCCESS	<input type="button" value="Job actions"/>

Starting in v2.5+: run the Grouper container with the "daemon" argument or the appropriate env vars set ([see v2.5 container documentation](#))

In v5.8+ you can unschedule a job to stop it. If the job is hung on a network call (e.g. a database lock), you cannot stop it, but if it is processing data it will stop shortly. You can do a few jstacks on the daemon process to see if it is making progress.

See this document to have a [unix service](#) which controls the loader

Note: it is also possible to run the loader in another webapp if you like

The long-term roadmap is to have the ability to run the loader in a web services instance or UI instance

See Also

For [Grouper v2.5+](#) see [this page on Daemon configuration](#).

For Grouper Daemon configuration options, see the [Daemon section of the Grouper API](#) page.

For info on pruning Daemon logs, see [Ongoing Maintenance Tasks](#)