admin/setting/tasklogging

- Managing scheduled tasks |]
- Colours |]
- Task processing |]
- Format for scheduling tasks |]
- Fail delay |]
- Running individual tasks |]
- Run scheduled tasks ASAP and asynchronously on server |]
- Ad hoc tasks |]
- Tasks running now |]
- Launching a task from CLI |]

Server settings

- System paths
- Session handling
- HTTP
- Maintenance mode
- Cleanup
- Environment
- Site registration
- Mail configuration
- OAuth 2 clients

- OAuth 2 services
- Scheduled tasks

- 1 Managing scheduled tasks
- 2 Colours
- 3 Task processing
- 4 Format for scheduling tasks
- 5 Fail delay
- 6 Running individual tasks
- 7 Run scheduled tasks ASAP and asynchronously on server
- 8 Ad hoc tasks
- 9 Tasks running now
- 10 Launching a task from CLI

Managing scheduled tasks |]

An administrator can schedule routine tasks very precisely from *Site administration > Server > Tasks > Scheduled tasks.*

Note that you still need to run the cron scripts (admin/cli/cron.php or https://yoursite/admin/cron.php) at regular intervals. It is recommended that the cron is run every minute.

Clicking the edit icon allows the administrator to specify the minute/hour/day/month or day of the week the task is to be run. It is also possible to reset the task to its default setting or disable it completely.

The column 'Next run' provides information on whether a plugin or a task is disabled (as well as the date that a task will next run).

Colours |]

If you see a scheduled task shaded yellow/orange this means the setting has been changed from default.

If you see a scheduled task shaded red, this means it has reached its maximum retry, i.e. it is almost certainly not working.

scheduledtaskcolours.png
Scheduled task colours

Task processing |]

From *Site administration* > *Server* > *Tasks* > *Task processing*, you can disable/enable cron and view and modify the processing of scheduled and ad hoc tasks.

The length of time to keep polling for additional tasks can be managed from the *Keep alive* setting (with a default value of 3 minutes.)

Format for scheduling tasks |]

When typing into the fields, the format is the same as for Unix cron. Examples are as follows and are according to which field you are editing:

```
* is every minute, hour, day, month
*/2 is every two minutes, every two hours or every second day
2-10 is every minute between two and ten past the hour or every hour between 2 and 10 am
0 is every Sunday
1 is every Monday or every January
2,5 is the second and 5th of the month, or February and May, or Tuesday and Friday.
```

Default: R means that a random value is picked when saving changes. This helps to distribute tasks more equally instead of running too many of them on the hour, midnight etc.

Fail delay |]

If you're trying to debug a cron task, you may notice the Fail Delay becomes populated with a number. This is the time in seconds the cron will delay running the task. To sidestep this problem for development purposes, take a look at the Scheduled Tasks section here: Administration via command line#Scheduled tasks.

Running individual tasks |]

To be able to run individual scheduled tasks via 'Run now' links on the scheduled tasks page, 'Allow 'Run now' for scheduled tasks' (tool_task | enablerunnow) in Site administration / Security / Site security settings should be enabled AND 'Path to PHP CLI' (pathtophp) in *Site administration > Server > System paths* should be set.

> examplescheduledtasks.png
> Scheduled tasks with 'Run now' option.

Run scheduled tasks ASAP and asynchronously on server |]

New feature

in **Moodle 5.3**!

Admins can now choose to run a scheduled task in the background (asynchronously) via cron, instead of running it directly in a browser.

Before this change, the only way to manually trigger a scheduled task was to click ‘Run now’ in Site administration > Server > Scheduled tasks. This ran the task in the browser, which could time out if the task took a long time to complete.

Now, admins can choose ‘Run ASAP’, shown with ‘Run now’ option for any task that isn't already scheduled to run immediately. After confirming the action message, the task is queued to run in the background the next time cron runs, so there is no risk of a browser timeout.

A success notification confirms the task has been queued, and ASAP is displayed in the ‘Next run’ column for the task until cron picks it up.

> Run scheduled tasks ASAP and asynchronously on server
> Run scheduled tasks ASAP and asynchronously on server

Ad hoc tasks |]

From *Site administration > Server > Tasks > Ad hoc tasks* you can see ad hoc tasks grouped by class name.

Clicking on a component/class name you then have a link to run that particular ad hoc task from the UI.

Ad hoc tasks can be also be run individually from the command line via admin/cli/adhoc_task.php. Options are:

- Run by id
- By class
- Only failed

It is also possible to limit the number of tasks to run, for example no more than four at a time.

See MDL-70975 for example CLI commands to run the above ad hoc tasks.

Tasks running now |]

Currently running tasks may be viewed from *Site administration > Server > Tasks > Tasks running now*.

TasksRunningNow.png
Tasks running now

Launching a task from CLI |]

You can also launch individual task from Command Line Interface (see Administration via command line).

Source: docs.moodle.org/503/en/admin/setting/tasklogging — MoodleDocs (GPL) · Imported 2026-11-10.

□□□ #1
□ cocobook-sync □□□ 2026-11-10 15:49:41 UTC
□ cocobook-sync □□□ 2026-11-10 15:49:41 UTC