

Setting up a Scheduled Task for your Plugin

There are two parts to setting up a scheduled task in your plugin.

- Create the task class under `classes\task` with an appropriate name using the template shown.
- Define the task and default schedule in `db\tasks.php`

Creating the task class

First, create the directory structure `classes\task` in your plugin folder. You will need to create at least the `task` directory and perhaps the `classes` one as well.

Here's the basic template for a scheduled task using the example of a local plugin called `yourplugin` as a placeholder. This file would be saved as `example_task.php` to match the name of the class.

```
/**
 * {Example_task} class definition
 *
 * @package      local/yourplugin
 * @author       Your Name * @license      http://www.gnu.org/copyleft/gpl.html GNU GPL v3 or
later
 */
namespace local_yourplugin\task;
use core\task\scheduled_task;
class example_task extends scheduled_task {
    /**
     * Get scheduled task name.
     *
     * @return string
     * @throws \coding_exception
     */
}
```

```

    public function get_name() {        return get_string("examplename",
"local_yourplugin");
    }
    /**
     * Execute the scheduled task.
     */
    public function execute() {
        global $CFG;        require_once($CFG->dirroot .
'/local/yourplugin/locallib.php');
        local_yourplugin_execute_task();
    }
}

```

A few things to note about this code:

- You need an entry for `examplename` in your plugin's language pack that is the readable name of the task (used by `get_name()` method) and shown on the scheduled tasks page.
- The `execute()` method includes `$CFG` and includes the `locallib.php` for your plugin where you define the details of the `local_yourplugin_execute_task()` function.

You can change the `local_yourplugin_execute_task()` to anything you like. The goal is to define the logic for this function in the local library (`locallib.php`) and **not** in the task class.

Define the task schedule

The file `tasks.php` can then be created in the `db\` folder. Here's an example of how this is defined using the `$tasks` array. If your plugin has more than one scheduled task, simply add another child array.

```

/**
 * Schedule tasks
 *
 * @package      local/yourplugin
 * @author       Your Name * @license      http://www.gnu.org/copyleft/gpl.html GNU GPL v3 or
later
 */

$tasks = [

```

```
[
    'classname' => 'local_yourplugin\task\example_task',
    'blocking' => 0,
    'minute' => 0,
    'hour' => 6,
    'day' => '*',
    'month' => '*',
    'dayofweek' => '*',
    'disabled' => 1
]
```

In this example:

- The namespace to the class definition is used for the `classname` so make sure you have namespaced your task class accordingly.
- The default scheduled is to run every day at 6am
- The task is disabled by default

Revision #3

Created Thu, Aug 8, 2019 12:12 PM by [Moodle Wiki](#)

Updated Thu, Aug 8, 2019 12:31 PM by [Moodle Wiki](#)