

Cron Jobs

In the admin panel, it gives the warning: **Warning:** Changing settings on this page can have drastic effects if your server is not configured correctly. It is important that you **consult the user manual**, so that you may fully understand what is happening, before changing any settings on this page.

The **reason for the strong warning** is so you come to this page to learn what is needed to change certain settings on the page, specifically what you need to do before switching to *Option B*. Don't let the warning scare you too much though, if you are familiar with setting up cron tasks for your hosting (or your hosting has good instructions for doing so), you should be fine with setting up the basic heartbeat to run once a minute, just use the [Manually Run Heartbeat every minute](#) example listed in your admin panel for doing so. The main thing, is after you have set it up to use a manual cron task, after about a week or so, be sure to come back to the cron tasks page and look at the *Last Run* time for each of the cron tasks. If none of them have been run recently you will need to double check to make sure you have it all set up correctly, or just switch it back to doing *Option A*. It is a good idea to come back to the page periodically just to make sure everything is still running smoothly.

How cron tasks work...what a cron task is:

A cron task, with respect to this software, is an individual task that is meant to be run at a specific interval, to do things like closing listings, moving expired listings to the archive table, removing old images, etc.

By default, the software will run the cron tasks by calling the heartbeat every time someone views a page on the client side. The heartbeat can also be called using something called a "cron job", but this requires the ability to use cron jobs on your host. Adding a cron job is usually done through your hosting control panel, or by using telnet or ssh. If you are not sure how to add a cron job on your host, you can safely keep the settings to the default, and the cron tasks will be run automatically for you any time someone views your site.

The "heartbeat" is a back-end process that determines what tasks need to be run according to the interval for that specific task, and runs the tasks needed. For instance, the task of closing listings that have expired is scheduled to run every 60 seconds. If the heartbeat sees that the last time that task has been run was 75 seconds ago, it knows to run that task, but if it sees that it has only been 46 seconds since that task has run, it skips the task.

As we mentioned before, the heartbeat can be called automatically, or through the use of a cron job, but there is an alternative to using the heartbeat to run the tasks needed. Instead, you can call each cron task using a cron job that specifically runs a certain task or tasks. This would allow you to not rely on the intervals built into the software to run each cron task, but to set your own schedule for each cron task to be run. For instance, you could run the `archive_listings` cron task every 1st Monday of the month at 1AM if you wanted to. You are only limited by the limitations in how cron jobs are scheduled. How to create customized schedules using cron jobs is beyond the scope of this document, but there are plenty of tutorials on how to use cron jobs. Run a search for "cron help" in your favorite web search engine. Your host may also have special instructions that pertain specifically to your site, so consult any FAQ's or help documents provided by your hosting provider as well.

It is not recommended to change any of the settings on this page unless you are familiar with using

cron jobs, and are able to test to make sure everything is set up correctly. If something is mis-configured, it can keep things like closing listings from being run, so caution must be used if you do change the configuration on this page.

Cron tasks can either be run using the internal schedule for each task, using the "heartbeat", or can be manually run.

Cron Settings

Run "Heartbeat":

- a. Automatically when page loads (Default)
- b. Manually with Cron Job (Requires Manual Server-Side Cron Jobs)

Option a

Is recommended for most sites, as it requires no additional configuration on the server side. With this option, every page view on the client side uses javascript to call the heartbeat. Since it is running separately from the main part of the page view by using javascript to call it, the heartbeat will not slow down the page load time.

Option b

Is for advanced users only, as it requires additional configuration on the server side to set up cron jobs. Using this option turns off the automatic heartbeat on every client side page view, and relies on the heartbeat to be called using a "cron job" set up to run however often you set it to. If you do decide to set up a cron job to run the heartbeat, we recommend setting it to run in 1-5 minute intervals.

If you wish to run each task manually instead of letting the heartbeat decide which tasks to run, you would also select option b to turn off automatic calls to the heartbeat routine.

Cron Security Key:

This is the key that must be used when calling specific tasks to be run using a cron routine, but it is not required when calling the heartbeat routine. It is randomly generated the first time the cron back-end is used, but you can change it if you wish, or clear the field to have the key reset to a new random value.

Time before lock removed (Deadlock prevention):

When a task is being called it is "locked" by the system to prevent the task from being run multiple

times at once. For example, this process helps to prevent pitfalls such as multiple e-mails being sent.

The "Time before lock removed" setting, is the amount of time, in seconds, before such a lock is manually removed by the system. This is done to prevent a situation known as "deadlock", where if a task "dies" because of an error, it stays locked even though the task is technically not running any more. If you are not sure what this setting is used for, keep the setting at the default of 1800 Seconds (30 minutes).

Cron Task Information

Available Tasks/Heartbeat Schedule Info:

Lists all of the tasks that are currently in the system, and lists information about each task. Addons can add their own tasks, you can tell it is an addon because the task name will have a ":" in the task name. If you were to install and enable the example addon version 1.1 or higher, you would see a new task named "example:example_task".

If you are going to run specific tasks using cron jobs (instead of using the built-in heartbeat task), the task name must match one of the tasks listed in this section in your admin, or that task will not be run.

Cron Job Examples:

This section gives examples of cron jobs you may set up, with the site URL and cron key set to match your site.

Manually Run Heartbeat every minute

This is an example of how you would use a cron job to call the heartbeat task, instead of loading the heartbeat task automatically on each page load. If your server does not have the command line browser "lynx", you must substitute whatever command line browser your server does have. For example you might substitute "lynx" with "wget" or something similar.

Run the made-up tasks task1 and task2 every Monday at 1AM

This is an example of how to set it up to manually run certain tasks using a cron job, instead of relying on the heartbeat to run the tasks needed. In the example, it uses **task1** and **task2**, which are **not real task names**, this is just an example. The [first example](#) of running the heartbeat would actually work on servers that have lynx, where this example would do nothing since the tasks are not real.

SSH Command Line Options

Using "command line php" to run cron tasks: In the first 2 examples, using lynx in the cron job will

work as long as lynx is installed on your server, or if your server has another command line browser that can be used instead.

If your server does not have a command line browser, it is also set up to be called using PHP to run it directly on the command line, however this is not guaranteed to work since server configurations can vary widely. If you wish to test using PHP instead of lynx, wget, or another method, using SSH try the following command:

```
php /path/to/cron.php --help
```

Replace "**/path/to/cron.php**" with the absolute path to the file cron.php on your server, which will be listed in your admin. This will display information on how to use cron.php to be run on the command line, without using a browser.

From:

<https://geodesicsolutions.org/wiki/> - Geodesic Solutions Community Wiki

Permanent link:

https://geodesicsolutions.org/wiki/admin_menu/site_setup/cron_jobs/start?rev=1244562011

Last update: **2014/09/25 16:55**

