

Setting Up Scheduled Sending Using Cron

Cron is a schedule based service that runs on just about every UNIX and Linux server. It allows you to run commands and scripts at a certain time and date that you can set in a "crontab". Cron can be set to run any Perl or PHP script and basically any command that can be run from a command line.

In this tutorial I will explain exactly what cron is and how you can use it to schedule the execution of the system's scheduled sending script (so you can schedule your email campaigns to be sent at a later date) on your web server.

I will also show you briefly how to create a crontab in Plesk Reloaded and in Cpanel (Xskin) using the online control panels. I will be using the system as the application that I am setting up the cron job for. The actual scheduled sending script will be used to send out emails and autoresponders.

Getting Started

In order to get started with cron, you must have a server that is currently running cron. Your host will also be able to tell you where you are able to setup crontabs -- whether it's from a web based control panel, or through a telnet terminal window.

Cron - Paint by Colors

Lets explore and begin to understand the crontab command and then we'll be able to apply it using the method that we have available (via control panel or telnet). Look at the following basic crontab:

```
* * * * * /bin/php -q /home/user/public_html/folder1/script.php
```

A typical crontab command, explained below.

I'll color code each "section" to allow me to visually explain the different parts of this crontab.

```
* * * * * /bin/php -q /home/user/public_html/folder1/script.php
```

A color coded crontab command.

Here's an explanation of the above cron command:

Minute of the hour

Set to a number between 0 – 59 for the corresponding minute of the hour or * for every minute

Hour of the day

Set to a number between 0-23 for the corresponding hour of the day or * for every hour

Day of the month

Set to number between 1-31 for the corresponding day of the month or * for every day

Month of the year

Set to a number between 1-12 for the corresponding month of the year or * for every month You can use the name of the month of the year as well eg. march (not case sensitive)

Day of the week

Set to a number between 0-7 for the corresponding day of the week (0 or 7 is Sunday) You can use the name of the day of the week as well. Eg. Tuesday (not case sensitive)

Beginning of the command line

Here you specify the path to PHP or Perl and any flags you might need.

The path to your script

Specify the path to your script that you want to run.

What is the asterisk and what does it do?

The Asterisk specifies that the script should be run on every occurrence of a time change. For example, if you want to set a cron to run every hour, then you would place an asterisk in the 'hour' column. Eg. 0 * 1 1 1 Setting a range within a column.

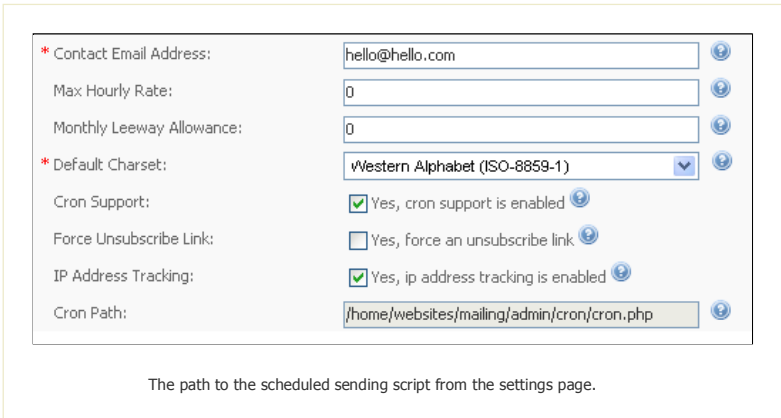
You can set a range of values in the cron command so that it will run within that range. Eg. 2-5 will run 2, 3, 4 and 5 of the column you set it in. In the day of the month column, for example, this 2-5 setting will run on the 2nd, 3rd, 4th and 5th of the month and will look like this. 0 * 2-5 * * Setting the frequency within each column.

If you wanted to run a cronjob every 5 minutes, you would be able to use */5 in the minutes column. If it was every 5 hours, then */5 could be placed in the hours column. Many would confuse this */5 as a fraction, but it is actually setting the frequency that the cron takes place in that column. It means 'every five' rather than a fifth of it.

The command line

This section can be any command that can be run from a command line. In this case we are setting it up to run a script.

- Path to PHP – UNIX and Linux have a special variable called **env php** which tells you where PHP is installed on your server. We'll use this instead of trying to figure out the path to PHP manually.
- Flags for output – In this example I've used the `-q` flag which runs cron in quiet mode; this flag stops some unnecessary messages. You can also add `/dev/null` after the server path to stop error messages being output as well.
- Server Path to your script – You will need to specify the full server path directly to your script. You can find this on the settings page, as shown below:



The path to the scheduled sending script from the settings page.

Setting up cron to run the scheduling script

Now that we know what the different parts of the command line actually do, we can begin to setup a cron job for a real use. In the system, we need to run the `send_emails.php` file (for sending newsletters) and the `send_autos.php` file (for autoresponders) in the "users" folder of the application. (Please see the documentation for exact file names as this may change with later versions of the system.)

- Decide what frequency you wish to run the script. For every hour on the hour, you would use `0 * * * *`
- Create the cronjob. Login to telnet or your control panel and type `crontab -e` at the command line. This will let you edit your crontab file. Type this text in:


```
0 * * * * env php -q /home/user1/public_html/sendstudio/admin/cron/cron.php
```
- Sit back and let cron do the rest.

Setting Crontabs Through Your Hosting Control Panel

Some 'front ends' to cron allow you to set the time and the command line for crontabs through your online hosting control panel. Look below for some examples of Cpanel and Plesk.

Cpanel 9 (Xskin)

Login to your control panel and look for the cronjobs icon There are two 'modes' to set cron jobs in Cpanel – you can choose either the standard or the advanced mode. The standard mode looks like this:



Entry 1

Command to run:

Minute(s):

- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24

Delete

Hour(s):

- Every Hour
- Every Other Hour
- Every Four Hours
- Every Six Hours
- 0 = 12 AM/Midnight

Months(s):

- Every Month
- January
- February
- March
- April

Day(s):

- Every Day
- 1
- 2
- 3
- 4

Weekday(s):

- Every Weekday
- Sunday
- Monday
- Tuesday
- Wednesday

Setting up a cronjob in Cpanel using the simple options.

This is a very simple way to add the frequency of the cron without needing to know what the asterisk or the numbers mean. In the command line you then add your path to PHP and the full path to your script. In the top field, you place the email address where the output will be sent – this is usually to alert you of errors etc.

For the advanced mode you will see the following:

Please enter an email address where the cron output will be sent:

Minute	Hour	Day	Month	Weekday	Command
<input type="text" value="*/20"/>	<input type="text" value="*"/>	<input type="text" value="*"/>	<input type="text" value="*"/>	<input type="text" value="*"/>	<input type="text" value="env php -q /home/user1/public_html/sendstudio/"/>

[\[Go Back \]](#)

Setting up a cronjob in Cpanel using the advanced options.


In this mode, you simply place the full command and then enter your frequency in the corresponding columns – they are clearly labeled so you can see which column are using as well.

Plesk (Reloaded)

Login to your Plesk control panel and look for the crontab icon. Crontabs are called 'tasks' in Plesk. In order to create a task click on the add new task icon. All current tasks will be shown in the tasks section for the specific user(in the dropdown).You can enter the email for cron output for this user in the email address field.

Crontab tasks ↑ Up Level

Tools

 [Add New Task](#)

Preferences

Show Crontab of

Send Crontab messages to address

Tasks

Crontab tasks (1)


✗ Remove Selected

S	M	H	DM	M	DW	Command	
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	* /20 * * * * env php -q /home/user1/public_html/sendstudio/admin/cron/cron.php	<input type="checkbox"/>

Setting up a cronjob in Plesk using the tasks section.

Here's the "Add New Task" screen:

Tools

 [Disable](#)

Task

Minute *

Hour *

Day of the Month *

Month *

Day of the Week *

Command *

* Required fields

The 'Add New Task' screen in Plesk's tasks section.

All fields are clearly marked so you know what time factor you are dealing with. Then just add your command to the command field and the task is complete.