

Identifying the php.ini used for in command line/cron engine

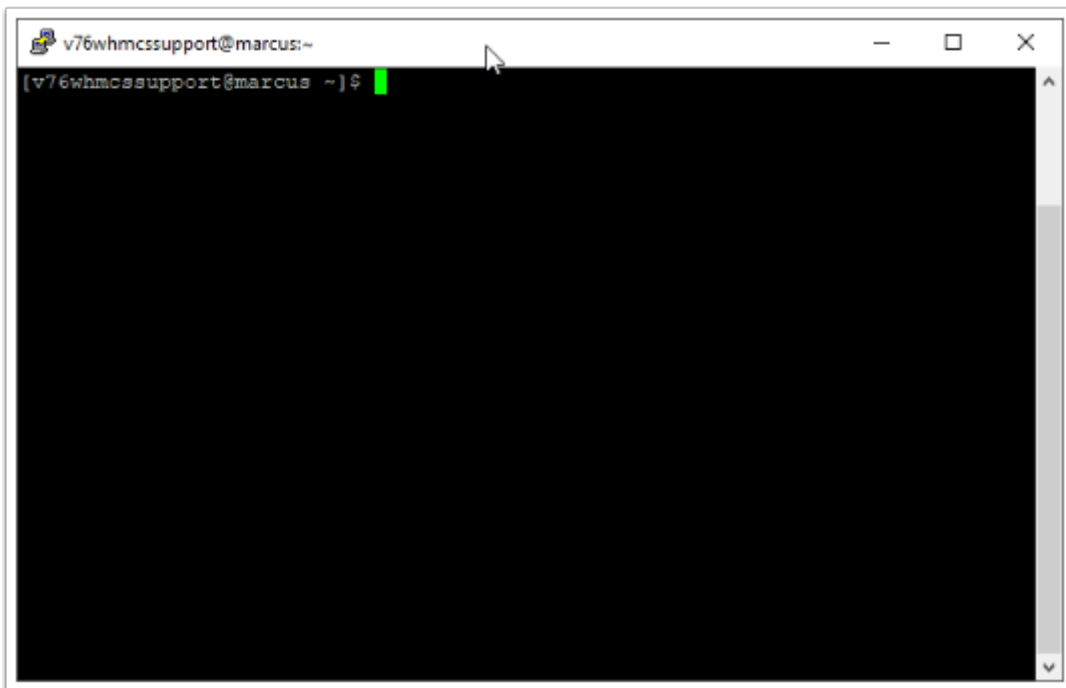
Many server configurations utilize a separate *php.ini* file in the command line or cron engine than the web server uses. This can sometimes cause issues with having different configurations. Fortunately there are ways to diagnose and troubleshoot this.

Identifying the php.ini file used via SSH

The first method to identify which *php.ini* file is being loaded by your cron engine is to use an SSH session.

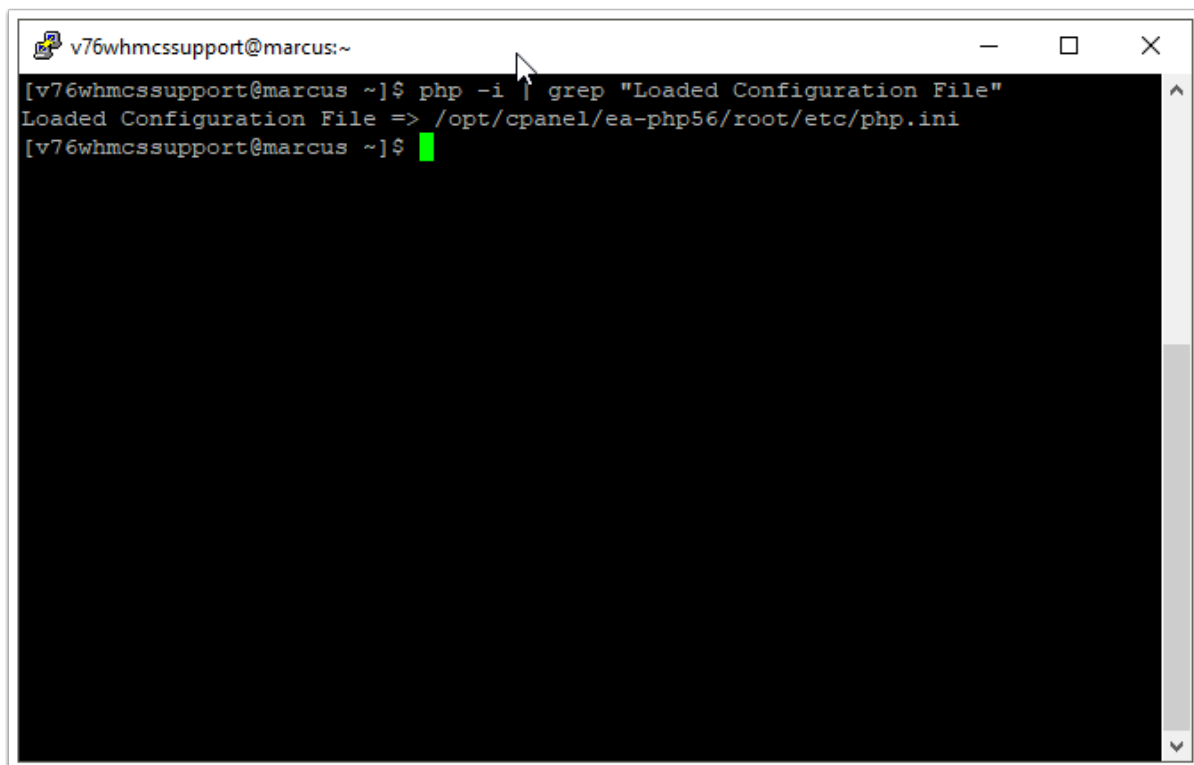
Using a computer-based terminal:

Open your terminal of choice and login to your server via SSH. Ideally, you will want to login with the same user account as your webserver uses, or that your WHMCS installation uses. It is best to avoid using *root* during troubleshooting as a properly configured server does not utilize the root user to run the web server or cron processes.



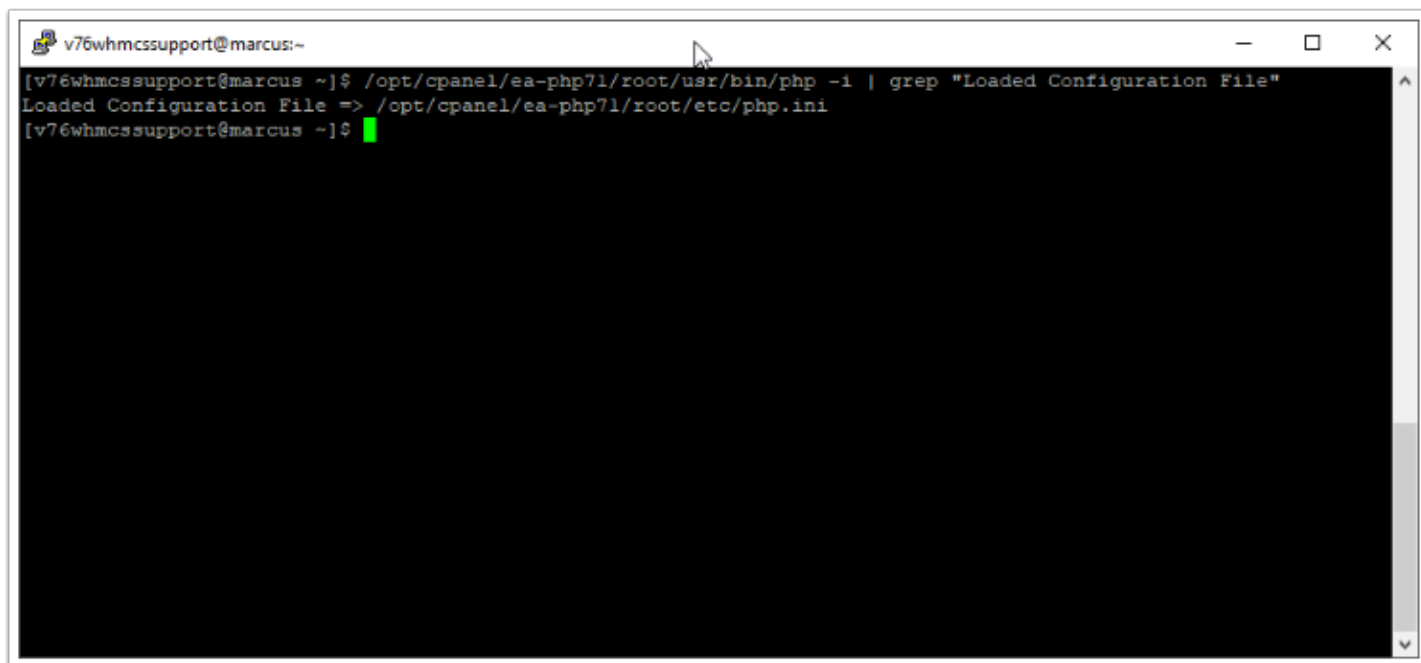
Once logged into the server, simply execute the following command:

```
php -i | grep "Loaded Configuration File"
```



```
v76whmcsupport@marcus:~  
[v76whmcsupport@marcus ~]$ php -i | grep "Loaded Configuration File"  
Loaded Configuration File => /opt/cpanel/ea-php56/root/etc/php.ini  
[v76whmcsupport@marcus ~]$
```

Doing this will show the configuration file that is being loaded when executing the *php* command. If you are forcing a specific version of PHP, or using the complete path you would simply substitute that for the *php* variable used above:

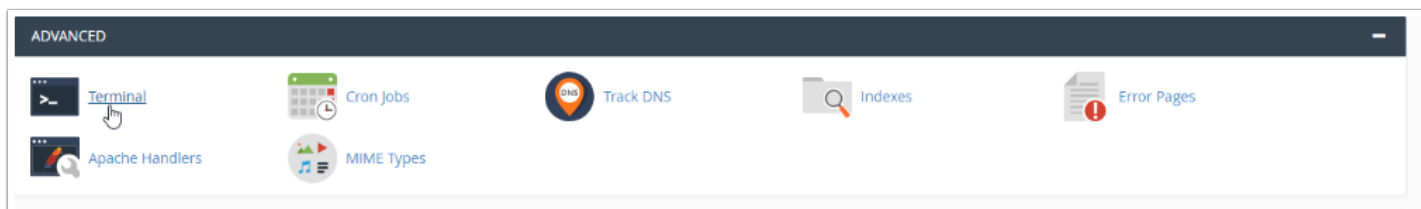


```
v76whmcsupport@marcus:~  
[v76whmcsupport@marcus ~]$ /opt/cpanel/ea-php71/root/usr/bin/php -i | grep "Loaded Configuration File"  
Loaded Configuration File => /opt/cpanel/ea-php71/root/etc/php.ini  
[v76whmcsupport@marcus ~]$
```

Using the cPanel based terminal:

Another option to check this is via the Terminal option within cPanel. This feature was added in cPanel & WHM Version 72. Various settings must be set in order for your account to have the Terminal option. You can review the cPanel Documentation for this feature on their [Documentation Site here](#).

Login to your cPanel and choose the *Terminal* option from the *Advanced* section.

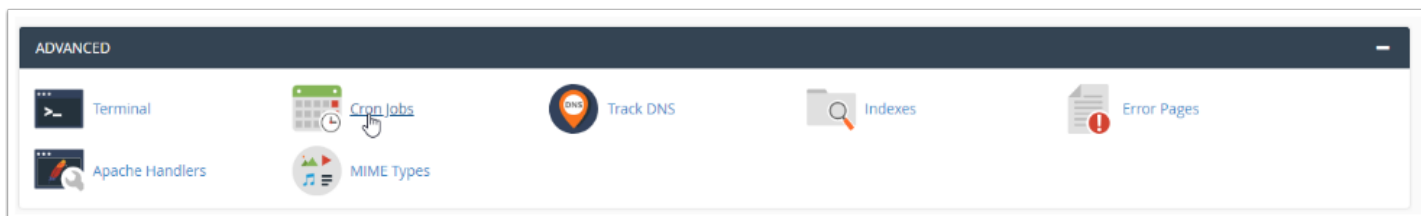


Once the terminal is opened, the steps are identical to using the Computer-Based Terminal above.

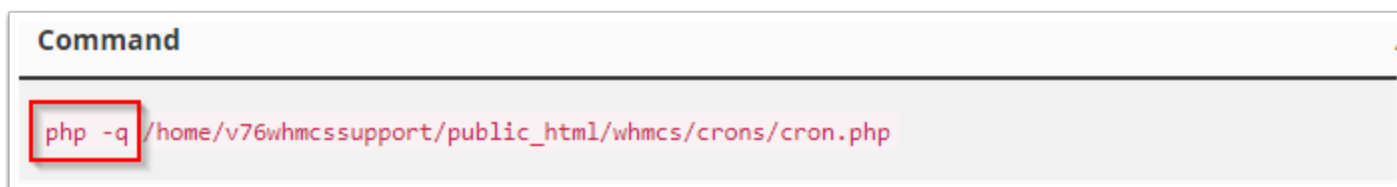
Using the Cron Job

One final option to check is via the cron engine itself. You can modify, or create, a cron job task to put this output to a file which can be read. For this example, we will be using cPanel.

First, Login to your cPanel and navigate to the *Cron Jobs* area, found under the *Advanced* section.



Once the cron job section is loaded, make note of the PHP path or variable being used by your existing WHMCS cronjob, as this is the path we will want to follow for troubleshooting.



This next step you have 2 options. You can modify the existing cronjob, or you can create a new one which is the preferred method. We will use the second option.

We will now create a new cronjob that runs every minute with a command similar to the one below:

```
php -i | grep "Loaded Configuration File" > /home/v76whmcssupport/public_html/whmcs/crons/phpoutput.txt
```

In this example, to make the file easy to find we are placing the output file in the same folder as the crons directory. In this case you would set that path to your crons directory path

Common Settings

Once Per Minute(*****)

Minute:

* Once Per Minute(*)

Hour:

* Every Hour (*)

Day:

* Every Day (*)

Month:

* Every Month (*)

Weekday:

* Every Day (*)

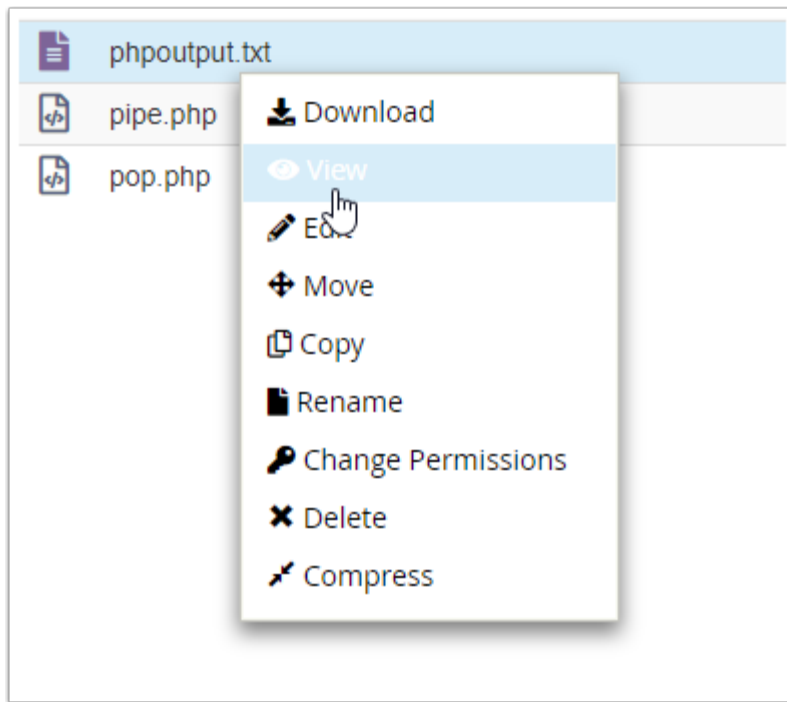
Command:

php -i | grep "Loaded Configuration File" > /home/v76whmcssupport/public_html/whmcs/crons/phpou

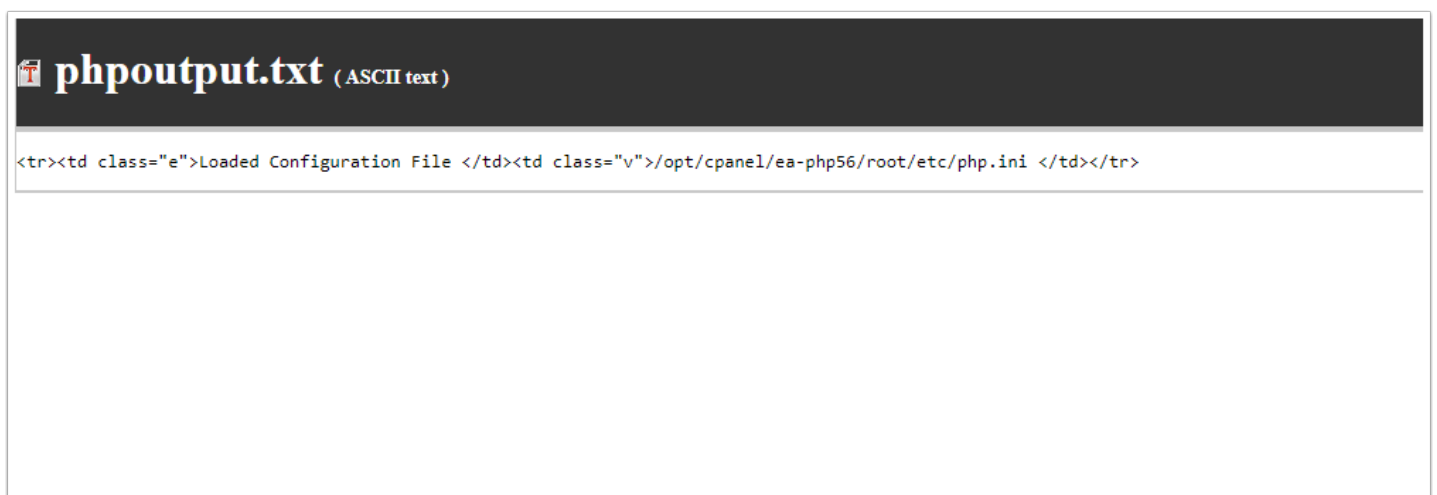
Add New Cron Job

We now want to head to the crons directory to look at the file output of the cron we just created. We will do this via the File Explorer in cPanel, but you could also check via FTP.

Note: The cron is setup to run 1 time per minute, so you must wait at least 1 minute (60 seconds) before the cron will be invoked or the output file will not be there.



We can now see the output by right clicking the file and choosing view:



In this example a PHP 5.6 environment is being used for the cron job.

After viewing the output, you should now remove the temporary cronjob to eliminate this unnecessary task from running.

What do I do with this information?

When looking for the proper PHP.ini file to edit, it is typically because settings need to be modified to satisfy the [WHMCS System Requirements](#). You will now need to work with your SysAdmin or Web Host to properly edit the PHP.ini to ensure all of the requirements are met, which should resolve any cron issues!

Resolving a Cron PHP Version Mismatch Warning


Once the php.ini file being used by the cron on your system has been identified, it can be compared against that used by the web server:

1. Navigate to **Utilities > System > PHP Info**
2. Examine the *Loaded Configuration File* value:

Server API	CGI/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/opt/cpanel/ea-php73/root/etc
Loaded Configuration File	/opt/cpanel/ea-php73/root/etc/php.ini
Scan this dir for additional .ini files	/opt/cpanel/ea-php73/root/etc:/opt/cpanel/ea-php73/root/etc/php.d:

3. In this example the loaded PHP configuration is located at */opt/cpanel/ea-php73/root/etc/php.ini*

This indicates PHP 7.3 is being used for the web server environment.

 In this example the *cron job* is using **PHP 5.6**. Whilst the *web server* is using **PHP 7.3**
Such version differences can cause unexpected behavior and errors in WHMCS

4. Work with your server admin/hosting provider to harmonise the PHP version used in both locations.

One option is to adjust the cron job command to use the exact same php.ini file as the web server. Eg:

```
php -c /path/to/php.ini -q /path/to/crons/cron.php
```

Continuing our example, this might produce a cron job command which looks something like:

```
php -c /opt/cpanel/ea-php73/root/etc/php.ini -q /home/v76whmcssupport/public_html/whmcs/crons/cron.php
```

Your server admin/hosting provider will be best placed to advise upon the most appropriate solution for your particular server environment and configuration as well as the file paths to use. We recommend consulting with them on any changes beforehand.