

# Nagios XI - Optimizing The PHP Settings File

Article Number: 611 | Rating: 1/5 from 1 votes | Last Updated: Thu, Jul 25, 2019 at 4:39 PM

## Overview

This KB article provides steps on optimizing the PHP settings file `php.ini` on your Nagios XI server. There are certain situations where the default settings in PHP need increasing to of Nagios XI on larger systems.

PHP is the scripting language that is used to provide the Nagios XI web pages.

## Editing Files

In many steps of this article you will be required to edit files. This documentation will use the `vi` text editor. When using the `vi` editor:

- To make changes press `i` on the keyboard first to enter insert mode
- Press `Esc` to exit insert mode
- When you have finished, save the changes in `vi` by typing `:wq` and press Enter

## php.ini File Location

To determine the location of your `php.ini` file execute the following command:

```
find /etc -name php.ini
```

If there are multiple results then the one in the `apache` directory is the one that needs changing.

## Restarting Apache Service

In this KB article, after making a change to the `php.ini` file you will need to restart the Apache service using one of the commands below:

### RHEL 6 | CentOS 6 | Oracle Linux 6

```
service httpd restart
```

### RHEL 7 | CentOS 7 | Oracle Linux 7

```
systemctl restart httpd.service
```

### Ubuntu 14

```
service apache2 restart
```

### Debian | Ubuntu 16/18

```
systemctl restart apache2.service
```

## max\_input\_vars

This is the most common setting that needs adjusting. There is a default limit of 1000 that exists to prevent denial of service attacks that can occur.

When viewing the Apache `httpd` error log:

```
tail /var/log/httpd/*error_log
```

You will see an event like the following:

```
[Tue Jun 30 18:16:34.132636 2015] [!error] [pid 16785] [client 10.25.254.50:28694] PHP Warning: Unknown: Input variables exceeded 1000.  
To increase the limit change max_input_vars in php.ini.
```

To increase the value, edit the `php.ini` file and check to see if the option already exists in the `php.ini` file by typing this command in `vi`:

```
/max_input_vars
```

If the option does not exist the following message will be displayed:

```
E486: Pattern not found: max_input_vars
```

If the option does not exist then it's a simple matter of adding the line to the end of the file. Press **Shift + G** to jump to the end of the file.

If the option does exist then it will take you directly to that line which you can adjust.

Define the limit to a larger number like 50000:

```
max_input_vars = 50000
```

After making the changes save the file and then [restart the httpd service](#).

Once the service has been restarted go and check to see if this has resolved your problem. If it has not resolved the problem, check the logs again to see if you are still hitting a limit steps above to increase the value to a larger number like 10000.

## memory\_limit

---

In larger Nagios XI deployments the amount of memory that PHP requires may exceed the default limit of 128M. This setting exists to prevent a poorly written script from eating up all server.

When viewing the Apache httpd error log:

```
tail /var/log/httpd/*error_log
```

You will see an event like the following:

```
[Tue Jul 12 02:22:07 2016] [error] [client 10.25.254.50] PHP Fatal error: Allowed memory size of 134217728 bytes exhausted (tried to allocate 2048 bytes) in /usr/local/nagiosxi/html/backend/includes/xml2json.php on line 243
```

To increase the value, edit the `php.ini` file and check to see if the option already exists in the `php.ini` file by typing this command in vi:

```
/memory_limit
```

If the option does not exist the following message will be displayed:

```
E486: Pattern not found: memory_limit
```

If the option does not exist then it's a simple matter of adding the line to the end of the file. Press **Shift + G** to jump to the end of the file.

If the option does exist then it will take you directly to that line which you can adjust.

Define the limit to a larger number like 1024M:

```
memory_limit = 1024M
```

After making the changes save the file and then [restart the httpd service](#).

Once the service has been restarted go and check to see if this has resolved your problem. If it has not resolved the problem, check the logs again to see if you are still hitting a limit steps above to increase the value to a larger number like 2048M.

## max\_execution\_time

---

In larger Nagios XI deployments the PHP engine may take a while to execute. There is a default limit of 30 seconds that exists to prevent a poorly written script from locking up the server.

When viewing the Apache httpd error log:

```
tail /var/log/httpd/*error_log
```

You will see an event like the following:

```
Fatal error: Maximum execution time of 30 seconds exceeded
```

To increase the value, edit the `php.ini` file and check to see if the option already exists in the `php.ini` file by typing this command in vi:

```
/max_execution_time
```

If the option does not exist the following message will be displayed:

```
E486: Pattern not found: max_execution_time
```

If the option does not exist then it's a simple matter of adding the line to the end of the file. Press **Shift + G** to jump to the end of the file.

If the option does exist then it will take you directly to that line which you can adjust.

Define the limit to a larger number like 90:

```
max_execution_time = 90
```

After making the changes save the file and then [restart the httpd service](#).

Once the service has been restarted go and check to see if this has resolved your problem. If it has not resolved the problem, check the logs again to see if you are still hitting a limit steps above to increase the value to a larger number like 180.

## max\_input\_time

---

In larger Nagios XI deployments the PHP engine may take a while to execute. The default setting is -1, this means that the value from the `max_execution_time` setting is used instead to prevent a poorly written script from locking up the server.

When viewing the Apache httpd error log:

```
tail /var/log/httpd/*error_log
```

You will see an event like the following:

```
Fatal error: Maximum input time of 30 seconds exceeded
```

To increase the value, edit the `php.ini` file and check to see if the option already exists in the `php.ini` file by typing this command in vi:

```
/max_input_time
```

If the option does not exist the following message will be displayed:

```
E486: Pattern not found: max_input_time
```

If the option does not exist then it's a simple matter of adding the line to the end of the file. Press **Shift + G** to jump to the end of the file.

If the option does exist then it will take you directly to that line which you can adjust.

Define the limit to a larger number like 90:

```
max_input_time = 90
```

After making the changes save the file and then [restart the httpd service](#).

Once the service has been restarted go and check to see if this has resolved your problem. If it has not resolved the problem, check the logs again to see if you are still hitting a limit steps above to increase the value to a larger number like 180.

## Additional Resources

---

The online documentation for PHP explains these settings in more detail, please refer to the following links:

[Description of core php.ini directives](#)

[Runtime Configuration](#)

## Final Thoughts

---

For any support related questions please visit the [Nagios Support Forums](#) at:

<http://support.nagios.com/forum/>

Posted by: **tlea** - Thu, Jul 13, 2017 at 9:23 PM. This article has been viewed 4094 times.

Online URL: <https://support.nagios.com/kb/article/nagios-xi-optimizing-the-php-settings-file-611.html>