

# Delete old IBM i Apache log files automatically

## Issue

By default, Apache on the IBM i creates new log files every day. After a while, there can be a lot of these old log files in the directory, making it difficult to page down to the current files when needed. Also, these old files are just taking up space, so we may as well be rid of them.

## Environment

IBM i, any version.

## Resolution

The Apache variant for IBM i includes some nice directives that help make keeping up with the Apache logs easy. One directive, LogCycle, specifies that Apache logs should be detached periodically and replaced by new ones. LogCycle states how often this should happen with one of the values Off (never), Hourly, Daily, Weekly, or Monthly. If LogCycle is not specified, the default value of Daily is used. The configuration we ship with Zend Server for IBM i does not specify LogCycle, so new Apache logs are spawned daily.

This is nice, but the log file directory `/www/zendphp74/logs` for versions Zend Server 2020.x and higher, `/www/zendphp74/logs` for versions **Zend Server 9.x, Zend Server 2018.x and Zend Server 2019.x** and `/www/zendsvr6/logs` for versions **6 - 8.5.x** tends to fill up with these things at the rate of two a day (error and access logs), so after a few months, there are quite a lot of them.

This can be a nuisance when researching an Apache issue, scrolling through all of the old logs, or even just waiting for the re-sort by new date in Navigator. And besides, old Apache logs are pretty useless anyway, because we almost never need to go back more than a few days when looking into an Apache issue. It is pretty easy to get rid of the old files manually by just deleting them out of Navigator or using the `wrklnc` command, but it would be a pain to have to do that on some regular basis, and anyway, you have much better things to be doing with your time.

Fortunately, IBM Apache also provides the LogMaint directive. This directive gets rid of old files based on either how many days old the file is, or how much space all the files take up cumulatively. Most of the time we don't really worry too much about how big the files get, we just don't want to have too many of them. So for this example we will get rid of them after a specified number of days.

**Note: LogMaint won't delete old files if LogCycle is set to Off.**

Obviously, because if LogCycle is off, there is only one log file.

LogMaint only has three parameters, for the path, the number of days for expiration, and aggregate size limit.

The path can be relative to the server path in Apache, which in our case is something like `'www/zendphp74'` (for versions 2020.x and higher) or `'www/zendphp7'` (for versions 9.x, 2018.x and 2019.x) or `'www/zendsvr6'` (for versions 6 - 8.5.x).

So the directory `'/www/zendphp74/logs'` (or `'/www/zendphp7/logs'` or `'/www/zendsvr6/logs'`) becomes just `'logs'` in the path parameter.

The file names include a suffix for the date, something like `'access_log.Q115122700'`, where `'Q115122700'` means file number 00 written on Dec 27, 2015. In some cases you can write more than one file in a day, which is why there are an extra two digits at the end. The path parameter just needs the name of the file without any suffix, so all files with this name and any suffix are evaluated. So, the path parameter for the access logs is just `'logs/access_log'` (no quotes needed in the directive).

The expired directive just tells how many days old a file can be and still be allowed to stay around. This is just an integer. For example, `'10'` means a file ten days old at evaluation time will be kept, anything eleven days or older will be deleted.

The size limit directive is an integer number specifying the maximum aggregate size in bytes of all files specified by the path, excluding the log file currently in use. So if all the sizes of all the files in the specified path exceed this many bytes, files will be deleted, oldest first, until the total size is less than the limit size. Setting this value to zero means there is no size limit.

So, putting them all together, the LogMaint directive for access logs to be kept for ten days with no maximum aggregate size looks like this:

```
LogMaint logs/access_log 10 0
```

It is important to remember that Apache has to be running for the log maintenance to be performed. Log maintenance is performed once a day, at a specified time. If Apache is not running at that time, there will be no log maintenance done that day. By default, the log maintenance time is midnight. For some shops, this might be a time when Apache is down for backups, an IPL, or some other routine reason. In this case, the LogMaintHour directive can be set to run the log maintenance at a time when Apache will be running.

LogMaintHour has one parameter which is the hour of the time to run the maintenance. The maintenance is always done at the beginning of the specified hour. The parameter is a number in the range of 0 to 23, where 0 represents midnight and 23 represents 11 p.m. So, to run log maintenance every morning at 3:00

```
LogMaintHour 3
```

There are three logs we like to maintain. They are the access log, the error log, and the FastCGI errors. FastCGI errors are not daily log files. There is one file for each error. This is OK, they can still be deleted by age. So, putting them all together, here are the lines to add:

```
## Keep access logs 10 days, error logs 10 days, FastCGI logs 10 days
LogMaint logs/access_log 10 0
LogMaint logs/error_log 10 0
LogMaint logs/error_zfcgi 10 0

# Maintain Logs at 3 am (0 = midnight, 23 = 11 pm, etc)
# Set for an hour when the server is active (i.e. not during an IPL or backup, for example)
LogMaintHour 3
```

These lines can be added to the configuration file for the Apache instance.

The configuration file for Zend Server versions 2020.x and higher is:

```
/www/zendphp74/conf/httpd.conf
```

The configuration file for Zend Server versions 9.x, 2018.x and 2019:

```
/www/zendphp7/conf/httpd.conf
```

The configuration file for Zend Server versions 6 - 8.5.x is:

```
/www/zendsvr6/conf/httpd.conf
```



**Note: Please back up this file before editing it.**

It is always important to back up any configuration file before editing, because mistakes in the file can prevent start up.

The above lines can be placed anywhere in the file, and in fact you will find some comments at the end of the file that were added as a guide. However, I like to put them closer to the top, near the other logging directives. Usually, I put them right after this line:

```
CustomLog logs/access_log combined
```

After adding your lines to the httpd.conf file, save it and restart Apache for this change to take effect.

## References

The directives discussed in this article can be found in the IBM Online documentation here:

[IBM Knowledge Center - Module core](#)

Another useful tip that can help control the size of your access log can be found here:

[How to exclude Apache Access log entries](#)