

# EasyApache FAQ

## General questions

### What is EasyApache?

EasyApache is software that installs, modifies, and validates your Apache web server, PHP, Tomcat, and other components of your web server. WHM's *EasyApache 3* interface (*Home >> Software >> EasyApache 3*) provides an interface that allows you to select which components you want EasyApache to install.

You do not have to use EasyApache, but it provides a convenient and easy method to modify your web server. Your cPanel & WHM license includes the EasyApache software.

For more information about EasyApache, read our [EasyApache](#) documentation.

### Why should I run EasyApache, and how often should I run it?

EasyApache provides the following benefits:

- Automatic updates to PHP, Apache, and the modules that you select.
- Decreased security vulnerabilities due to the automatic updates.
- Access to configure Apache and PHP via the WHM or command line interfaces.
- A simplified method to add, remove, or install components of your web server.
- Recommendations about compatibility.
- Easy access to information about the options you select.
- Minimal downtime due to an automatic recovery process. If EasyApache detects an error in the server configuration and it is unable to automatically correct it, then EasyApache will revert your server to the previously working configuration.

We recommend that you run EasyApache whenever there is an update to software that you use. For example, if you use PHP on your server, run EasyApache each time that PHP releases an update.

### How do I run EasyApache?

#### Notes:

- Before you run EasyApache, we recommend that you update your server to the latest cPanel & WHM version. WHM will perform a check of your version of cPanel & WHM, and will prompt you to update if you are not on the latest version. This will allow EasyApache to configure and install the latest available features.
  - To update your cPanel & WHM version, use WHM's [Upgrade to Latest Version](#) interface (*Home >> cPanel >> Upgrade to Latest Version*).
  - To update your cPanel & WHM version via the command line, run the `/scripts/upcp` script as the `root` user.
- When you run EasyApache, it will automatically update to the latest available version of EasyApache.

To run EasyApache via the WHM interface, use WHM's *EasyApache 3* interface (*Home >> Software >> EasyApache 3*).

To run EasyApache via the command line, run the `/scripts/easyapache` script as the `root` user.

### What do EXPERIMENTAL, DEPRECATED, END OF LIFE (EOL), and NOT SUPPORTED mean?

When you configure EasyApache, the following terms in the WHM interface describe a product or feature's state of development:

- **EXPERIMENTAL** — This product is new and not fully tested. It may have compatibility issues with other products.
- **DEPRECATED** — The product is considered out-of-date, but we will still support it. We may provide patches for compatibility and security issues, but you may wish to upgrade to a newer version soon.
- **END OF LIFE** — This product has reached End of Life (EOL). We provide the product in its current state and will not release security patches for it. We strongly encourage you to upgrade to a newer version.
- **NOT SUPPORTED** — EasyApache does not support this product. This product will not receive further updates.

### How can I run EasyApache and keep my custom configuration?

If you manually modified your Apache, PHP or Tomcat configurations, we recommend that you perform the following actions before you run

EasyApache:

1. Read the [EasyApache - Development](#) documentation.
2. Access the *EasyApache 3* interface and verify that EasyApache has the versions of the software that you use. EasyApache will not make changes to your server until you direct it to initiate the build process.
3. Save a backup copy of all of your configuration files.
4. Run EasyApache with the options that you wish to use.
5. Use the procedures in our [EasyApache - Development](#) documentation to modify the configuration files to restore the necessary parts of your custom configuration.

**Warning:**

EasyApache will automatically remove changes to the configuration if they are not in the correct files. Read our [EasyApache - Development](#) documentation for more information.

## Which EasyApache options should I select?

The EasyApache options that you select will determine what EasyApache builds into your web server. As a general rule, do not select an option unless you need it. For each option that you select, make certain that you understand the functionality of the option and any security vulnerabilities that may come with it.

**Important:**

We **strongly** recommend that you do not use options marked *DEPRECATED* or *NOT SUPPORTED*. They may introduce security vulnerabilities or instability to your server.

## What should I do if I need an Apache module, PHP module, or PHP version that is not available in EasyApache?

EasyApache provides a mechanism that allows you to add options to EasyApache via custom modules. For example, you can add PHP or Apache modules, or PHP versions.

For more information on custom modules, read our [Custom Modules](#) documentation.

## What is the Apache `FollowSymLinks` option?

This option allows Apache to serve files which are located inside of the virtual host's document root. For example:

```
ln -s /home/bob/public_html/original.txt /home/bob/public_html/link.txt
```

## What is the Apache `SymLinksIfOwnerMatch` option?

If you enable symlinks via `FollowSymLinks`, Apache will only serve content via a symlink when the owner of the symlink matches the owner of the file (For example, the user `bob` links to a file that `bob` owns). However, [Apache's documentation for symlinks](#) states that a potential race condition with symlinks exists and the directive will affect server performance.

## What is the race condition?

Apache performs a test for symlinks as it turns a URL into a file path on disk. Apache handlers expect to operate on the file path rather than on an open file handle. So, a race condition exists titled *time of check time of use* (TOCTOU) between the step where Apache validates the path and when a content handler accesses the symlinked file.

## What does the Rack911 patch do?

The Rack911 patch turns the Apache `FollowSymLinks` directive into the Apache `'FollowSymLinks' + 'SymLinksIfOwnerMatch'` directive. This provides a quick way to activate both of these directives in the `httpd.conf` file at the same time.

## How does the Rack911 patch protect my website from the race condition?

The Rack911 patch itself does not protect against the race condition. Instead, this patch changes the Apache `Options FollowSymlinks` directive into a shortcut for the Apache `Options FollowSymlinks SymLinksIfOwnerMatch` directive. If you use this patch, your users **cannot** turn off both the Apache `SymLinksIfOwnerMatch` directive and `symlinks`. Because Apache does not provide a detailed ACL approach for these options, this solution provides a quick way to enable and disable symlinks at a system-wide level within the `httpd.conf` file.

## What does the new Bluehost.com patch do?

The Bluehost.com patch modifies Apache and APR so that Apache cannot access certain files. However, the Bluehost.com patch only affects requests that Apache's default handler processes (for example, static files such as `.html` files). The Bluehost patch does not affect requests that application content handlers such as `mod_php` process.

Traditionally, systems administrators assign content handlers to the content that they want to serve. For example, the `mod_php` handler serves PHP scripts, and the `mod_cgi` handler serves arbitrary CGI scripts. When Apache cannot find a content handler associated with an incoming request, Apache resorts to the `default_handler`. The `default_handler` reads in the contents of a file and displays it as-is to the requestor (for example, `cat file.txt > requestor`).

A problem occurs when an application runs as the `nobody` user. The application will create files on the server that the user `nobody` owns, which does not match the owner. Because the patch prevents access to static content that does not match the owner, the user cannot view these files.

## Why would I want to use the Bluehost.com patch if the files it creates are inaccessible?

If your applications run as a specific user, then the files the application writes will possess the correct ownership.

## If the Bluehost.com patch prevents access to files that the user does not own, how does it determine what the user owns?

The Bluehost.com patch compares the UID of the requested file to the owner of the document root. If the owners do not match, Apache returns the following error code:

```
HTTP code 404; Not Found error.
```

Finally, the system writes a warning to the `/usr/local/apache/error_log` file that will resemble the following example:

```
[Wed Sep 13 10:45:31.596881 2017] [core:notice] [pid 1207:tid 139670931613824] AH00094: Command line: '/usr/sbin/httpd'
```

## Common issues and errors

When EasyApache builds, it creates a log file that contains all of the EasyApache process's output. EasyApache stores the build log files in the `/usr/local/cpanel/logs/easy/apache` directory.

If EasyApache fails to build successfully, EasyApache provides a general statement of the error near the end of the log file, and then a more specific failure error. If you see any of the following text in the EasyApache log file, look further down in the log file to determine the cause:

- Could not ensure pkglist
- Syntax error on line ...
- Failed to generate a syntactically correct Apache configuration
- Configuration problem detected on line ... of file ...

## When I try to run EasyApache, I get the error "EasyApache is currently running"

This error message indicates that your system detects another instance of EasyApache and will not proceed until that instance completes. You cannot run multiple instances of EasyApache at the same time. Some common causes of this error include the following situations:

- EasyApache is open in another browser tab. The issue will resolve after the other instance of EasyApache completes.
- Something initiated the EasyApache script (`/scripts/easyapache`) to run via the command line. The issue will resolve after the other instance of EasyApache completes.
- The EasyApache process hung and cannot complete. The system administrator will need to investigate the cause of the issue. We

recommend that you determine the cause of the issue before you kill the EasyApache process.

## When I try to run EasyApache, it repeatedly says that it is trying to update

The first thing that EasyApache does is check to see if there are any updates available. The most common causes of this issue relate to network connectivity. The EasyApache software requires access to our update servers, and will retry repeatedly if it cannot reach them.

In rarer cases, you may experience this issue when we are in the process of an EasyApache release. If you are certain that there is no connectivity issue, then we recommend that you wait up to one hour and then run EasyApache to receive the update.

## EasyApache completed successfully, but now some sites that use PHP load a blank page or a PHP error

The most common cause of this issue is that the new EasyApache configuration uses a different PHP version than the previous EasyApache configuration used.

The EasyApache build may have changed your PHP version from one minor version to another. For example, code written for PHP version 5.3 may not be compatible with PHP version 5.4. EasyApache will not automatically change the version unless you select a default profile, or if your last EasyApache build contained a version of PHP that we no longer support.

**Note:**

Updates to minor revisions of PHP do not usually affect compatibility. For example, code written for PHP version 5.4.28 should still function correctly with PHP version 5.4.29.

To resolve the issue, select the correct PHP version and rebuild EasyApache. If the version of PHP is not available, the developer of the incompatible code will need to either modify the code or you will need to use a custom module for the necessary PHP version.

**Warning:**

We **strongly** recommend that you do not use a version of PHP that has reached End of Life (EOL) or that we mark as *DEPRECATED* or *NOT SUPPORTED*. They may introduce security vulnerabilities or instability to your server.

This issue may also happen if you do not select a PHP handler after you run EasyApache. Use WHM's [Configure PHP and suEXEC](#) interface (*Home >> Service Configuration >> Configure PHP and suEXEC*) to select a PHP handler.

## EasyApache completed successfully, but now some or all sites load an Apache error

If you include options that have different file permissions or security settings than your previous build, you may need to make additional adjustments. For example, the modules suPHP and ModRuid2 require specific file permissions to work correctly.

Before you can resolve the issue, you will need to determine what the root cause of the issue is. If the Apache error that the web browser displays is not clear, review the following sources of information:

- The Apache error log — You can access the Apache error log for your sites in the [Raw Apache Log Download](#) interface (*Home >> Account Functions >> Raw Apache Log Download*), which may help you determine the cause of the specific error.
- The EasyApache build log — The EasyApache build log will help you determine if there was an error in the build that was not a complete failure so the EasyApache build process completed. To access your EasyApache build log, click the *Build Log Manager* link in the *EasyApache 3* interface (*Home >> Software >> EasyApache 3*).
- The EasyApache build history — The EasyApache build history may help you determine what changed in your most recent EasyApache build. To access your EasyApache build history, click the *Build History* link in the *EasyApache 3* interface.
- Your `.htaccess` files — For information about `.htaccess` files, view Apache's [Apache HTTP Server Tutorial: .htaccess files](#) document ation for Apache version 2.4, or Apache's [Apache HTTPD Server Tutorial: .htaccess files](#) documentation for Apache version 2.2.

## EasyApache fails with an error that includes "cannot open shared object file: No such file or directory"

The full error that you receive will look similar to the following text, but the file names in the error may be different:

```
Syntax error on line 2 of /usr/local/apache/conf/includes/pre_main_2.conf:
Cannot load /usr/local/apache/modules/mod_fcgid.so into server:
/usr/local/apache/modules/mod_fcgid.so: cannot open shared object file: No
such file or directory
```

This error message means that EasyApache was unable to locate, access, or load the file that is in the error. The following issues can cause this error:

- A typo in the directory or file name.
- If the module is a custom module, you may have extracted it to a different directory.
- Incorrect permissions on the file.
- The module depends on libraries that Apache is unable to load.

To resolve the error message, correct all locations where the issue occurs in the Apache configuration.

## EasyApache fails with an error about ModSecurity

The ModSecurity software firewall can cause a variety of different errors if the configuration is not correct.

For example, the following error indicates that a rule in the configuration uses a *DEPRECATED* ModSecurity feature.

```
ModSecurity: WARNING Using transformations in SecDefaultAction is
deprecated (/usr/local/apache/conf/turtle-rules/10_asl_antimalware.conf: 25
).
```

The following error indicates that the configuration has two or more rules with the same ID:

```
Syntax error on line 176 of
/usr/local/apache/conf/turtle-rules/10_asl_rules.conf:
ModSecurity: Found another rule with the same id
```

For more information on ModSecurity and how to correct issues with your rules, read our [Apache Module: ModSecurity](#) documentation and the [modsecurity](#) website.

## EasyApache fails with the error "Error: Protected multilib versions"

This error indicates the following situation:

- EasyApache requires a specific version of a package that is on your server.
- Your server currently has a different version of that package.
- EasyApache is unable to update the package because it is part of a multiple library (*multilib*) package that it cannot modify.

To correct this issue, the system administrator needs to update the *multilib* package to the version that EasyApache requires. In some cases, the command `yum update` will resolve the issue.

## EasyApache fails with a configuration problem and specifies a syntax error such as "Either all Options must start with + or -, or no Option may"

The actual error message that you receive may vary. The following is an example error message that is related to the `SymLinksIfOwnerMatch` Apache option:

```
Configuration problem detected on line 67 of file
/usr/local/apache/conf/httpd.conf.xxxxxxxxxx: Either all Options must start
with + or -, or no Option may.
```

```
--- /usr/local/apache/conf/httpd.conf.xxxxxxxxxx ---
61ScriptAlias /mailman /usr/local/cpanel/3rdparty/mailman/cgi-bin/
62ScriptAlias /scgi-bin /usr/local/cpanel/cgi-sys/scgiwrap
63
64
65<Directory "/">
66     AllowOverride All
67 ===>     Options ExecCGI FollowSymLinks Includes IncludesNOEXEC
-Indexes -MultiViews SymLinksIfOwnerMatch <===
68</Directory>
69
70<Directory "/usr/local/apache/htdocs">
71     Options All
72     AllowOverride None
73     Require all granted
--- /usr/local/apache/conf/httpd.conf.xxxxxxxxxx ---
```

To resolve this issue, correct the options that are in your Apache configuration files. For more information on the syntax and options that are available for your Apache configuration, view Apache's [Directive Quick Reference](#) documentation for Apache version 2.2, or Apache's [Directive Quick Reference](#) documentation for Apache version 2.4.

## EasyApache fails with the error "Skipping rest of tests: httpd not running"

This error indicates that after EasyApache completes the build process, the Apache process (`httpd`) did not start correctly. This may happen if you install additional web server software, such as Litespeed. If Litespeed is the cause of the issue, the full error message will include the following output:

```
Skipping rest of tests: httpd not running

[last system cmd output]
root      12924  0.5  0.2  32524 10172 ?          S    02:17   0:00 litespeed
(lshttpd)
root      12927  0.0  0.0   6156   500 ?          S    02:17   0:00 httpd
(lscgid)
nobody    12928  0.2  0.2 118780 10180 ?        Sl   02:17   0:00 litespeed
(lshttpd)
root      13280  0.0  0.0  23200  1764 ?          S    02:17   0:00
/usr/local/cpanel/3rdparty/bin/perl /usr/local/cpanel/scripts/syscmdrc
1397776656.27340 ps uxawww | grep http
root      13281  0.0  0.0  11336  1176 ?          S    02:17   0:00 sh -c ps
uxawww | grep http
root      13283  0.0  0.0   6392   692 ?          S    02:17   0:00 grep http
```

To resolve this issue, perform the following steps:

1. Kill the Litespeed processes.

2. Run EasyApache.
3. After the EasyApache build completes, kill the Apache processes.
4. Start Litespeed.

## EasyApache fails with the error "[Errno 14] PYCURL ERROR 22 - "The requested URL returned error: 404 Not Found" Trying other mirror"

This error indicates that the yum repository list on your server contains a URL that does not resolve to a server correctly. If the other entries in your yum repository list work correctly, this should not cause EasyApache to fail.

To resolve the issue, correct the issues in your yum repository list. For more information on your yum repository list, read your operating system documentation.

## EasyApache fails with the error "Cannot allocate memory"

This error indicates that your server does not have enough memory to run EasyApache. The more options that you select in EasyApache, the more memory EasyApache requires to build successfully. You **must** increase the available memory on the server to at least 512 MB of RAM before you run EasyApache.

## EasyApache fails with the error "Error unpacking rpm package"

This error indicates that your server was unable to unpack an RPM that EasyApache requires. The full error message may look similar to the following example:

```
Error unpacking rpm package coreutils-8.4-31.el6_5.1.x86_64

error: unpacking of archive failed on file /bin/ln: cpio: rename
```

This may be due to an issue with the package that is in the error. In some cases, the `yum update` command will resolve the issue.

## EasyApache fails with a disk space error such as "No space left on device"

A disk space error indicates that your server does not have the disk space to unpack all of the packages that EasyApache requires. The specific error message you receive may vary. The following is an example of a disk space error message:

```
Running Transaction Test
Transaction Test Succeeded
Running Transaction

  Updating      : glibc-common-2.12-1.132.el6_5.1.x86_64
1/14Error in <unknown> scriptlet in rpm package
glibc-common-2.12-1.132.el6_5.1.x86_64

error: error creating temporary file /var/tmp/rpm-tmp.6IuFNg: No space left
on device
error: Couldn't create temporary file for
%triggerin(glibc-common-2.12-1.132.el6_5.1.x86_64): No space left on device
```

To resolve the issue, the system administrator **must** increase the amount of available disk space or inodes on the server.

## EasyApache fails with the error "The server's system package manager, 'YUM', failed."

This error indicates that the yum update system experienced an error. Many problems could cause this issue.

For example, yum may fail for any of the following reasons:

- Multilib version problems.
- A corrupt RPM database.
- Broken RPM dependencies.
- A large number of duplicate RPMs.
- Bad mirrors.
- Failed installations.
- RPMs could not add or remove files in system paths, due to a variety of conditions.

```
!! The server's system package manager, 'YUM', failed. !!
!!
This is the command that failed: yum -y install automake19 gettext
libstdc++.x86_64 libpng-devel readline-devel java-1.7.0-openjdk-devel
openssl libpng-dev zlib-devel autoconf261 libidn-devel gmake libidn libXpm
openssl-devel automake coreutils patch libltdl3-devel libltdl
libopenssl0.9.7-static-devel readline-dev libtool-ltdl-devel sed
libXpm-devel libXpm-dev lsof krb5-dev flex glibc-devel expat-dev krb5-devel
libstdc++-devel.x64_64 log4j xorg-x11-devel eclipse-ecj
jakarta-commons-collections libtool-ltdl libssl-dev pam-devel
libopenssl0-devel zlib-devel jakarta-commons-logging liblve-devel
expat-devel libopenssl0-dev gcc-c++ expat glibc-devel zlib bison
jakarta-commons-dbcplibjpeg-devel jakarta-commons-pool
libtool-libltdl-devel libtool openssl-dev libopenssl0 libz-devel
libjpeg-dev jakarta-taglibs-standard pam-dev fileutils libltdl-devel
libopenssl0.9.7-devel e2fsprogs-devel ca_root_nss make
libstdc++-dev.x86_64 libX11-devel libstdc++-devel.x86_64 gd cpp
xorg-x11-dev gcc ssl-dev lex autoconf

!!
!!
Since EasyApache was unable to resolve it automatically you should:
    1) Manually run the failed YUM command (shown above) via SSH
    2) See if your particular error is addressed at
http://go.cpanel.net/eaerror
    3) Resolve the YUM problem manually
    4) Re-run EasyApache

!!
!! Please visit http://go.cpanel.net/eaerror for help with this error. !!
```

To troubleshoot this error, run the following command:

```
yum update
```

During the update process, yum checks all of your packages for issues. The error or errors that this command produces will identify the source of the problem.

**Note:**

The most common reason for this error is multilib problems with the libstdc++ package.