

# How to Enable Additional CGI Scripts

## Warnings:

- This document describes an unsupported workaround that we do **not** guarantee will work in the future.
- This workaround does **not** work on systems that run EasyApache 4. Use the [Global Configuration](#) section of WHM's [Apache Configuration](#) interface ([WHM >> Home >> Service Configuration >> Apache Configuration](#)) instead and select the following options in the *Directory "/" Options* section:
  - `ExecCGI`
  - `FollowSymLinks`
  - `Includes`
  - `IncludesNOEXEC`
  - `Indexes`
  - `SymLinksIfOwnerMatch`
- For security reasons, we **strongly** recommend that you do **not** use CGI scripts in directories other than the `cgi-bin` directory.
- **Only** perform the steps that this document describes if all other available solutions fail.
- We **strongly** encourage you to back up your server before you perform this action.

## Overview

Apache version 2.2 allows you to use CGI scripts in directories other than the `cgi-bin` directory. To prevent the use of CGI scripts in directories other than the `cgi-bin`, you **must** disable the ability of individual `.htaccess` files to override the server settings.

Apache version 2.4 prevents the use of CGI scripts in directories other than the `cgi-bin` directory. To use CGI scripts in other directories, you **must** enable the ability of individual `.htaccess` files to override the server settings.

## Change the CGI scripts' availability.

Click the tabs below for directions about how to prevent the use of CGI scripts, or restore the default settings.

[Apache version 2.2](#) [Apache version 2.4](#)

### Disable CGI through `.htaccess`

To disable CGI through `.htaccess`, perform the following steps:

1. Log in to your server as the `root` user via SSH.
2. Change to the `/usr/local/apache/conf/` directory.
3. Create a backup of your `httpd.conf` file. For example:

```
cp httpd.conf httpd-old.conf
```

#### Note:

In this example, `http-old.conf` represents the backup file's name.

4. Open your `http.conf` file with a text editor and locate the section of the file that resembles the following example:

```
<Directory "/">
    Options +ExecCGI +FollowSymLinks +Includes +IncludesNOEXEC
+Indexes -MultiViews +SymLinksIfOwnerMatch
    AllowOverride All
</Directory>
```

5. Change the `AllowOverride` line's value from `All` to `None`.
6. Run the distiller to update the settings. To do this, run the following command:

```
/usr/local/cpanel/bin/apache_conf_distiller --update
```

Your server now **only** allows CGI scripts to run in the `cgi-bin` directories.

**Remember:**

Do **not** use this method for systems that run EasyApache 4. Use the [Global Configuration](#) section of WHM's [Apache Configuration](#) interface ([WHM >> Home >> Service Configuration >> Apache Configuration >> Global Configuration](#)) instead.

### Enable CGI through `.htaccess`

To enable CGI through `.htaccess`, perform the following steps:

1. Log in to your server as the `root` user via SSH.
2. Change to the `/usr/local/apache/conf/` directory.
3. Create a backup of your `httpd.conf` file. For example:

```
cp httpd.conf httpd-old.conf
```

**Note:**

In this example, `httpd-old.conf` represents the backup file's name.

4. Open your `httpd.conf` file with a text editor and locate the section of the file that resembles the following example:

```
<Directory "/">
  Options +ExecCGI +FollowSymLinks +Includes +IncludesNOEXEC
+Indexes -MultiViews +SymLinksIfOwnerMatch
  AllowOverride None
</Directory>
```

5. Change the `AllowOverride` line's value from `None` to `All`.
6. Run the distiller to update the settings. To do this, run the following command:

```
/usr/local/cpanel/bin/apache_conf_distiller --update
```

Your server now allows CGI scripts to run in **all** directories.

### Restore the `.htaccess` files' original setting

To restore the `.htaccess` files to their original setting, perform the following steps:

1. Log in to your server via SSH as the `root` user.
2. Change to the `/usr/local/apache/conf/` directory.
3. Restore the original `httpd.conf` file. To do this, run the following command:

```
rm httpd.conf
mv httpd-old.conf httpd.conf
```

**Remember:**

In this example, `httpd-old.conf` represents the backup file's name.

4. Run the distiller to update the settings. To do this, run the following command.

```
/usr/local/cpanel/bin/apache_conf_distiller --update
```

You have restored the `.htaccess` files to their default setting.

## Additional documentation

Suggested documentation [For cPanel users](#) [For WHM users](#) [For developers](#)

- [How to Enable Additional CGI Scripts](#)
- [ProFTPD Configuration for Host Access Control](#)
- [The Removal of cgiemail and cgiecho](#)
- [How to Create Ruby Web Applications](#)
- [How to Install a Node.js Application](#)

## Content by label

There is no content with the specified labels



- [How to Enable Additional CGI Scripts](#)
- [ProFTPD Configuration for Host Access Control](#)
- [The Removal of cgiemail and cgiecho](#)
- [How to Create Ruby Web Applications](#)
- [How to Install a Node.js Application](#)
  
- [cPanel API 1 Modules - DenyIp](#)
- [cPanel API 1 Functions - DenyIp::deldenyip](#)
- [cPanel API 1 Functions - DenyIp::adddenyip](#)
- [cPanel API 1 Functions - Htaccess::setindex](#)
- [cPanel API 1 Functions - Htaccess::del\\_user](#)