

Custom Templates

[Overview](#)

[Custom templates that will apply to all virtual hosts](#)

[Custom templates that will apply to an individual virtual host](#)

Overview

Custom template files allow you to modify how cPanel & WHM builds virtualhost entries in the Apache configuration.

Warning:

Misconfiguration of the custom templates may cause Apache to generate a syntactically-invalid `httpd.conf` file. This results in a server that does not respond to HTTP requests. Use caution when you create custom templates, and remember to place any directives that rely on certain Apache modules within appropriate `<IfModule>` blocks.

Custom templates that will apply to all virtual hosts

Note:

You **cannot** use custom templates for a parked domain (alias). Parked domains do not use the same type of entries in your Apache configuration.

cPanel & WHM version 11.36+cPanel & WHM version 11.36 and earlier

To create custom template files that affect how cPanel & WHM builds entries for all virtual hosts, perform the following steps:

1. Create a copy of one or more of the following files:
 - **Apache 2.2 with SSL** — `/var/cpanel/templates/apache2_2/ssl_vhost.default`
 - **Apache 2.4 with SSL** — `/var/cpanel/templates/apache2_4/ssl_vhost.default`
 - **Apache 2.2 without SSL** — `/var/cpanel/templates/apache2_2/vhost.default`
 - **Apache 2.4 without SSL** — `/var/cpanel/templates/apache2_4/vhost.default`
2. Rename the copied file to one of the following filenames:
 - `vhost.local` — Use this filename if you copied the `vhost.default` file.
 - `ssl_vhost.local` — Use this filename if you copied the `ssl_vhost.default` file.
3. Edit the `*.local` files to make the desired changes to your virtual host configuration.

Warnings:

- This method affects all of your virtual hosts as cPanel & WHM will use any `.local` files instead of the original `.default` files.
- You will **lose** all custom configurations when you upgrade to cPanel & WHM version 68. We **strongly recommend** that you back up any `.local` template customizations **before** you upgrade.

Custom templates for new accounts in cPanel & WHM version 11.36 and earlier

To create a custom template for new accounts in cPanel & WHM version 11.36 or older, perform

Featured

documentation

EasyApache 3

[Introduction – FAQ](#)

[Change Log – Release Notes](#)

[Profiles – Customization](#)

Apache

[Apache Modules](#)

[Advanced Apache Configuration](#)

PHP

[PHP Options](#)

[PHP Handlers](#)

Tomcat

[Migrate from Tomcat 5.5 to 7](#)

[How to Deploy Java Applications](#)

Important:

EasyApache 3 does **not** support new installations of Tomcat.

As of cPanel & WHM version 76, EasyApache 4 now supports Tomcat 8.5. For more information, read our [Tomcat documentation](#).

the following steps:

1. Create a custom version of one of the following files:
 - **Apache 2.2 with SSL** — `/usr/local/cpanel/etc/httpdtemplates/apache2_2/ssldefault`
 - **Apache 2.4 with SSL** — `/usr/local/cpanel/etc/httpdtemplates/apache2_4/ssldefault`
 - **Apache 2.2 without SSL** — `/usr/local/cpanel/etc/httpdtemplates/apache2_2/default`
 - **Apache 2.4 without SSL** — `/usr/local/cpanel/etc/httpdtemplates/apache2_4/default`
2. Save the newly created file in a directory where nothing will overwrite the file. The `/scripts/upcp` script will overwrite the contents of the `/usr/local/cpanel/etc/httpdtemplates/` directory each time cPanel & WHM checks for updates.

Important:

You **must** use the original file name.

3. To apply your custom template, create a `postupcp` script that replaces the default file with your custom file. cPanel & WHM runs `postupcp` scripts automatically after every cPanel update.

Note:

For more information about scripts that run after the `/usr/local/cpanel/scripts/upcp` script completes, read our [Script Hooks](#) documentation.

Custom templates that will apply to an individual virtual host

The `/var/cpanel/userdata/` directory

Warning:

We **strongly** recommend that you use include files to modify the content of an individual virtual host. For more information on how to use include files, read our [Modify Virtualhost Containers With Include Files](#) documentation.

cPanel & WHM stores information about domains in the subdirectories in the `/var/cpanel/userdata/` directory. cPanel & WHM then uses that data to build the virtual hosts for your Apache configuration.

Use the include files that allow you to modify an individual virtual host configuration. Alternatively, you can use a custom template that modifies the data in the `/var/cpanel/userdata/` directory for an individual domain.

The subdirectories within the `/var/cpanel/userdata/` directory contain YAML files. For more information about the YAML format, visit the [official YAML website](#).

Warning:

Do **not** manually edit these YAML files.

EasyApache loads these files into the `vhost` template variable. For more information about the `vhost` variable and Template Toolkit, visit the [Template Toolkit website](#).

How to create a custom template that will apply to a single virtual host

To create a custom template that affects how cPanel & WHM builds entries for only a single virtual host, perform the following steps:

1. Create a copy of one or more of the following files:
 - **Apache 2.2 with SSL** — `/var/cpanel/templates/apache2_2/ssl_vhost.default`
 - **Apache 2.4 with SSL** — `/var/cpanel/templates/apache2_4/ssl_vhost.default`
 - **Apache 2.2 without SSL** — `/var/cpanel/templates/apache2_2/vhost.default`
 - **Apache 2.4 without SSL** — `/var/cpanel/templates/apache2_4/vhost.default`

2. Rename the new copy of the file to one of the following filenames:
 - `vhost.local` — Use this filename if you copied the `vhost.default` file.
 - `ssl_vhost.local` — Use this filename if you copied the `ssl_vhost.default` file.
3. Edit the new file to make changes to your virtual host configuration.
4. Move the new file to any other directory.

Note:

You may move this file to any directory because you will specify the location of the file in the next step.

5. Use the following script to add the necessary information to the data structure:

```
perl -MYAML::Syck -e \  
'my $hr = YAML::Syck::LoadFile($ARGV[0]);$hr->{$ARGV[1]} = \  
$ARGV[2];YAML::Syck::DumpFile($ARGV[0],$hr);' \  
/var/cpanel/userdata/$username/$domainname \  
custom_vhost_template_ap2 $path/to/template
```