

Customize doctor diagnostic checks



- [doctor.yml format](#)
 - [Custom doctor.yml configuration files](#)
 - [Available check types](#)
-

Even though `wp doctor` comes with a [number of default diagnostic checks](#), it's designed with extensibility at its core. Checks are defined at runtime, read from a `doctor.yml` configuration file naming each check with its options.

doctor.yml format

Let's take a look at the first two checks in the included `doctor.yml`:

```
autoload-options-size:
  check: Autoload_Options_Size
constant-savequeries-falsy:
  check: Constant_Definition
  options:
    constant: SAVEQUERIES
    falsy: true
```

In this example:

- 'autoload-options-size' and 'constant-savequeries-falsy' are the *names* for the checks. Names must be unique amongst all registered checks.
- `Autoload_Options_Size` and `Constant_Definition` are reusable check classes in the `runcommand\Doctor\Checks` namespace. You can use them too, or you can write your own class extending `runcommand\Doctor\Checks\Check` and supply it as 'class: yourNamespace\yourClassName'.

- 'constant' and 'falsy' are configuration options accepted by the `Constant_Definition` class. In this case, we're telling doctor to ensure `SAVEQUERIES` is either false or undefined.

For the sake of completeness, it's also worth noting `Autoload_Options_Size` accepts 'threshold_kb' as an optional configuration option. The default value for 'threshold_kb' is 900, so it doesn't need to be included in the `doctor.yml`.

Custom doctor.yml configuration files

Run your own doctor checks by creating a `doctor.yml` and supplying it with `wp doctor check --config=doctor.yml`. Use different configurations for different environments by creating separate `prod.yml` and `dev.yml` files.

If you want your custom file to extend an existing doctor config, you can use the magical `_` config file option to define which config file to inherit. 'default' is a magic reference to the bundled `doctor.yml`; you also specify an entire file path.

Take a look at this example:

```
_:
  inherit: default
  skipped_checks:
    - autoload-options-size
constant-disallow-file-mods-falsy:
  check: Constant_Definition
  options:
    constant: DISALLOW_FILE_MODS
    falsy: true
plugin-akismet-active:
  check: Plugin_Status
  options:
    plugin: akismet
    status: active
plugin-akismet-valid-api-key:
  class: Akismet_Valid_API_Key
  require: akismet-valid-api-key.php
```

This custom `doctor.yml` file:

- Inherits all default diagnostic checks except for 'autoload-options-size'.
- Defines a 'constant-disallow-file-mods-falsy' check to ensure the `DISALLOW_FILE_MODS` constant is falsy.
- Defines a 'plugin-akismet-active' check to ensure Akismet is active.
- Defines a 'plugin-akismet-valid-api-key' [custom check in a akismet-valid-api-key.php file](#) to ensure Akismet has a valid API key.

Available check types

Some `wp doctor` check types are configurable, meaning the default setting can be changed, while other check types are abstracted in such a way that they can be reusable. For instance, the `Autoload_Options_Size` check accepts an option 'threshold_kb' while `Plugin_Status` accepts 'plugin' and 'status' as options.

The configurable check types include:

- `check: Autoload_Options_Size`: Accepts 'threshold_kb' as an option to set the threshold in kilobytes. Default value is 900.
- `check: Cron_Count`: Accepts 'threshold_count' as an option to set the threshold of total cron jobs. Default value is 50.
- `check: Cron_Duplicates`: Accepts 'threshold_count' as an option to set the threshold of duplicate cron jobs. Default value is 10.
- `check: Plugin_Active_Count`: Accepts 'threshold_count' as an option to set the threshold of total active plugins. Default is 80.
- `check: Plugin_Deactivated`: Accepts 'threshold_percentage' as an option to set the threshold of percentage deactivated plugins. Default is 40.

The abstracted check types include:

- `check: Constant_Definition`: Assert a given constant as defined, a specific value, or falsy. [Learn more](#).
- `check: File_Contents`: Check all or a selection of WordPress files for a given regex pattern. [Learn more](#).
- `check: Option_Value`: Assert a given option as a specific value. [Learn more](#).
- `check: Plugin_Status`: Assert a given plugin as active, installed, or uninstalled. [Learn more](#).

Of course, some check types don't need configuration options:

- `check: Core_Update`: Errors when new WordPress minor release is available; warns for major release.

- `check: Core_Verify_Checksums`: Verifies WordPress files against published checksums; errors on failure.
- `check: Plugin_Update`: Warns when there are plugin updates available.
- `check: Theme_Update`: Warns when there are theme updates available.

You can [write your own custom check type](#) by extending the `runcommand\Doctor\Checks\Check` class.

📄 #4

📄 cocobook-sync 📄 2025-06-25 18:05:32 UTC

📄 cocobook-sync 📄 2026-05-31 02:11:37 UTC