ignores

Sanity tests for individual files can be skipped, and specific errors can be ignored.

When to Ignore Errors

Sanity tests are designed to improve code quality and identify common issues with content. When issues are identified during development, those issues should be corrected.

As development of Ansible continues, sanity tests are expanded to detect issues that previous releases could not. To allow time for existing content to be updated to pass newer tests, ignore entries can be added. New content should not use ignores for existing sanity tests.

When code is fixed to resolve sanity test errors, any relevant ignores must also be removed. If the ignores are not removed, this will be reported as an unnecessary ignore error. This is intended to prevent future regressions due to the same error recurring after being fixed.

When to Skip Tests

Although rare, there are reasons for skipping a sanity test instead of ignoring the errors it reports.

If a sanity test results in a traceback when processing content, that error cannot be ignored. If this occurs, open a new bug report for the issue so it can be fixed. If the traceback occurs due to an issue with the content, that issue should be fixed. If the content is correct, the test will need to be skipped until the bug in the sanity test is fixed.

> Caution should be used when skipping sanity tests instead of ignoring them. Since the test is skipped entirely, resolution of the issue will not be automatically detected. This will prevent prevent regression detection from working once the issue has been resolved. For this reason it is a good idea to periodically review skipped entries manually to verify they are required.

Ignore File Location

The location of the ignore file depends on the type of content being tested.

Ansible Collections

Since sanity tests change between Ansible releases, a separate ignore file is needed for each Ansible major release.

The filename is `tests/sanity/ignore-X.Y.txt` where `X.Y` is the Ansible release being used to test the collection.

Maintaining a separate file for each Ansible release allows a collection to pass tests for multiple versions of Ansible.

Ansible

When testing Ansible, all ignores are placed in the `test/sanity/ignore.txt` file.

Only a single file is needed because `ansible-test` is developed and released as a part of Ansible itself.

Ignore File Format

The ignore file contains one entry per line. Each line consists of two columns, separated by a single space. Comments may be added at the end of an entry, started with a hash (`#`) character, which can be proceeded by zero or more spaces. Blank and comment only lines are not allowed.

The first column specifies the file path that the entry applies to. File paths must be relative to the root of the content being tested. This is either the Ansible source or an Ansible collection. File paths cannot contain a space or the hash (`#`) character.

The second column specifies the sanity test that the entry applies to. This will be the name of the sanity test. If the sanity test is specific to a version of Python, the name will include a dash (`-`) and the relevant Python version. If the named test uses error codes then the error code to ignore must be appended to the name of the test, separated by a colon (`:`).

Below are some example ignore entries for an Ansible collection:

```
roles/my_role/files/my_script.sh shellcheck:SC2154 # ignore undefined variable
plugins/modules/my_module.py validate-modules:E105 # ignore license check
plugins/modules/my_module.py import-3.8 # needs update to support collections.abc on Python 3.8+
```

It is also possible to skip a sanity test for a specific file. This is done by adding `!skip` after the sanity test name in the second column. When this is done, no error code is included, even if the sanity test uses error codes.

Below are some example skip entries for an Ansible collection:

```
plugins/module_utils/my_util.py validate-modules!skip # waiting for bug fix in module validator
```

```
plugins/lookup/my_plugin.py compile-2.6!skip # Python 2.6 is not supported on the controller
```

Ignore File Errors

There are various errors that can be reported for the ignore file itself:

- syntax errors parsing the ignore file
- references a file path that does not exist
- references to a sanity test that does not exist
- ignoring an error that does not occur
- ignoring a file which is skipped
- duplicate entries