**[Includes vs. Imports](http://docs.ansible.com/ansible/2.4/playbooks_reuse_includes.html" \l "id2)**

As noted in [Creating Reusable Playbooks](http://docs.ansible.com/ansible/2.4/playbooks_reuse.html), include and import statements are very similar, however the Ansible executor engine treats them very differently.

* All import\* statements are pre-processed at the time playbooks are parsed.
* All include\* statements are processed as they encountered during the execution of the playbook.
* New in version 2.4.

## [Importing Playbooks](http://docs.ansible.com/ansible/2.4/playbooks_reuse_includes.html#id3)

* It is possible to include playbooks inside a master playbook. For example:
* ---
* - import\_playbook: webservers.yml
* - import\_playbook: databases.yml
* The plays and tasks in each playbook listed will be run in the order they are listed, just as if they had been defined here directly.
* Prior to 2.4 only include was available and worked for both playbooks and tasks as both import and include.

## [Including and Importing Task Files](http://docs.ansible.com/ansible/2.4/playbooks_reuse_includes.html#id4)

Use of included task lists is a great way to define a role that system is going to fulfill. A task include file simply contains a flat list of tasks:

*# common\_tasks.yml*

---

- name: placeholder foo

command: /bin/foo

- name: placeholder bar

command: /bin/bar

You can then use import\_tasks or include\_tasks to include this file in your main task list:

tasks:

- import\_tasks: common\_tasks.yml

*# or*

- include\_tasks: common\_tasks.yml

You can also pass variables into imports and includes:

tasks:

- import\_tasks: wordpress.yml wp\_user=timmy

- import\_tasks: wordpress.yml wp\_user=alice

- import\_tasks: wordpress.yml wp\_user=bob

Variables can also be passed to include files using an alternative syntax, which also supports structured variables like dictionaries and lists:

tasks:

- include\_tasks: wordpress.yml

vars:

wp\_user: timmy

ssh\_keys:

- "**{{** lookup**(**'file'**,** 'keys/one.pub'**)** **}}**"

- "**{{** lookup**(**'file'**,** 'keys/two.pub'**)** **}}**"

Includes and imports can also be used in the handlers: section; for instance, if you want to define how to restart apache, you only have to do that once for all of your playbooks. You might make a handlers.yml that looks like:

*# more\_handlers.yml*

---

- name: restart apache

service: name=apache state=restarted

And in your main playbook file:

handlers:

- include\_tasks: more\_handlers.yml

*# or*

- import\_tasks: more\_handlers.yml

*main.yml*

- include: include\_outer.yml

with\_items:

- 1

- 2

- 3

*include\_outer.yml*

---

- debug:

msg: "Outer include: item={{ item }}"

- set\_fact:

outer\_loop: "{{ item }}"

- include: include\_inner.yml

vars:

inner\_item: "{{ item }}"

outer\_item: "{{ outer\_loop }}"

with\_items:

- A

- B

- C

*include\_inner.yml*

---

- debug:

msg: "Inner include: inner\_item={{ inner\_item }}, outer\_item={{ outer\_item }}"

The result:

"msg": "Outer include: item=1"

"msg": "Inner include: inner\_item=1, outer\_item=1"

"msg": "Inner include: inner\_item=1, outer\_item=1"

"msg": "Inner include: inner\_item=1, outer\_item=1"

"msg": "Outer include: item=2"

"msg": "Inner include: inner\_item=2, outer\_item=2"

"msg": "Inner include: inner\_item=2, outer\_item=2"

"msg": "Inner include: inner\_item=2, outer\_item=2"

"msg": "Outer include: item=3"

"msg": "Inner include: inner\_item=3, outer\_item=3"

"msg": "Inner include: inner\_item=3, outer\_item=3"

"msg": "Inner include: inner\_item=3, outer\_item=3"