

---

# **mcin.ensure\_dirs Documentation**

***Release 0.1.0***

**Christopher Steel**

December 22, 2016



<b>1</b>	<b>Usage Notes</b>	<b>3</b>
<b>2</b>	<b>Requirements</b>	<b>5</b>
<b>3</b>	<b>Role Variables</b>	<b>7</b>
<b>4</b>	<b>Dependencies</b>	<b>9</b>
<b>5</b>	<b>Example Playbook</b>	<b>11</b>
<b>6</b>	<b>License</b>	<b>13</b>
<b>7</b>	<b>Author Information</b>	<b>15</b>



The **ensure\_dirs** role is used as a dependancy to ensure for the creation or removal of directories on the local and/or remote system(s). The role also allows for setting the directory(ies) owner(s), group(s) and permissions and can be used to recursively ensure permissions.



---

### Usage Notes

---

When using a relative paths for remote directories the directory is created in, or is relative to, the ansible ssh users home directory.

When using with a relative path on the local system, for example “test/test1”, on the local system the director will be created in, or relative too, the directory in which the ansible-playbook command is run.





---

### Requirements

---

In most cases, except for perhaps testing, you would use this role as a dependancy for another role via that roles meta/main.yml **dependancies** [] option by passing one dictionary for the remote directories and / or a dictionary for any local directories you want to manipulate.



---

## Role Variables

---

**ensure\_dirs** uses two main variables in the form of Python dictionaries:

```
ensure_dirs_on_remote [] ensure_dirs_on_local []
```

The dependant role should create corresponding dictionary variables for one or both of the above variables. Generally these would be created in that roles **defaults** and/or **vars** directory.

For example, if you use the dependant roles default/main.yml you would describe the directories. In the example below the remote dictionary is called **another\_roles\_remote\_directories\_description**. In **another\_roles\_remote\_directories\_description** you can describe as many directories as desired.

At minimum each directory description must include a **name** and a **path**. If it is not specified, the **state** will be set to **directory**. Valid options for **state** are “**directory**” to ensure for the creation of a directory, or “**absent**” to ensure for the removal of a directory. See [Ansible - file - Sets attributes of files]([http://docs.ansible.com/ansible/file\\_module.html](http://docs.ansible.com/ansible/file_module.html)) for all possible options.

In our example **remote\_directory\_one** serves as a placeholder for our directory and **directory** the name and relative path of the directory. The placeholed **remote\_directory\_two** describes a directory using all the options currently supported for this role.

```
# file: roles/dependant_role/defaults/main.yml # # The my_remote_directories var contains the descrip-
tion of all of the # remote directories you want to ensure for #

another_roles_remote_directories_description:

# Directory definition using minimal amount of directory vars

remote_directory_one:

    state : “directory” path : “dir_01”

# Example using all possible directory variables:

remote_directory_two:

    state : “directory” path : “relative/subdir” owner : “bob” group : “sales” mode :
    “0750” recurse : “true”
```

```
### roles/dependant_role/meta/main.yml
```

In the dependant roles meta/main.yml we add our dependency(ies) and pass the Python dictionary(ies) that define(s) our our local and/or remote directory(ies). It might look something like this:

Note: In this example we have moved the dependencies variable to the top of the roles meta/main.yml file so that it is readily noticed.

```
### roles/dependant_role/defaults/main.yml example
```

In the dependant roles defaults directory we could use two groupings of variables, one for the directories we want to ensure for on remote systems and one grouping for directories we want on the local system where Ansible is being run.

— # file: roles/dependant\_role/defaults/main.yml # # \* to ensure for the removal of a directory set the state to be “absent” # \* to ensure for creation of directory set the state to be “directory” # \* The name or label of the collection of variables describing a directory, # for example **dir\_home\_test\_1**, should be unique. Other than that it # acts only as a place holder.

another\_roles\_remote\_directories\_description:

dir\_home\_test\_1:

state : “absent” path : “directory\_to\_remove” owner : “{{ ansible\_ssh\_user  
}}” group : “{{ ansible\_ssh\_user }}” mode : “0644”

dir\_home\_test\_2:

state : “directory” path : “creates\_a\_remote\_relative\_directory\_in\_the\_ansible\_ssh\_users\_home”  
owner : “{{ ansible\_ssh\_user }}” group : “{{ ansible\_ssh\_user }}” mode : “0644”

another\_roles\_local\_directories\_description:

dir\_home\_test\_3:

state : “absent” path : “relative\_to\_ansible\_project” owner : “{{ ansible\_ssh\_user  
}}” group : “{{ ansible\_ssh\_user }}” mode : “0644”

---

## Dependencies

---

None, **ensure\_dirs** is normally used as a dependency for other roles that require directory manipulations.



---

## Example Playbook

---

Under most circumstance this role does not require it's own playbook.





---

**License**

---

MIT



---

## Author Information

---

Christopher Steel christopher DOT steel AT mcgill DOT ca