

# Components

## Overview

Super groups have the ability to have their own components. The structure of a super group component is largely the same as a full CMS component with a few small differences. The primary difference related to the different application environments or types a full CMS component can support. While a full component may have controllers, views, and logic for site, admin, api, a super group component **only** has code for the equivalent of site.

Here's an example of CMS components vs super group components structure:

```
// CMS Component
/components
.. /com_example
.. .. /admin
.. .. .. /controllers
.. .. .. .. records.php
.. .. .. /language
.. .. .. .. /en-GB
.. .. .. .. .. en-GB.com_example.ini
.. .. .. /views
.. .. .. .. /records
.. .. .. .. .. /tmpl
.. .. .. .. .. .. display.php
.. .. .. example.php
.. .. /config
.. .. .. config.xml
.. .. /helpers
.. .. .. html.php
.. .. /models
.. .. .. record.php
.. .. /site
.. .. .. /controllers
.. .. .. .. one.php
.. .. .. /language
.. .. .. .. /en-GB
.. .. .. .. .. en-GB.com_example.ini
.. .. .. /views
.. .. .. .. /one
.. .. .. .. .. /tmpl
.. .. .. .. .. .. display.php'
.. .. .. example.php
.. .. .. router.php
.. .. example.xml
```

## COMPONENTS

---

```
// Super Group Component
/components
.. /com_example
.. .. /controllers
.. .. .. one.php
.. .. /helpers
.. .. .. html.php
.. .. /models
.. .. .. record.php
.. .. /language
.. .. .. /en-GB
.. .. .. .. en-GB.com_example.ini
.. .. /views
.. .. .. /one
.. .. .. .. /tmpl
.. .. .. .. .. display.php
.. .. example.php
.. .. example.xml
.. .. router.php
```

Here you can see that the contents of the super group component is the same as the /site sub-directory of the CMS component with the addition of the XML manifest and /helpers & /models directories. With include path updated, this covers the bulk of the changes. Views, models, routing, controllers, and so on should be handled in the same manner as full CMS components.

For more information regarding developing components see:

<https://hubzero.org/documentation/1.3.0/webdevs/components>

### Components Directory

```
{web_root}/site/groups/{group_id}/components/com_{component}/
```

### Component Language Files

```
{web_root}/site/groups/{group_id}/language/en-GB/en-GB.com_{component}.ini
```

### Component Paths

As a helper for super group component developers the path to the component directory is

## COMPONENTS

---

defined in a constant.

### JPATH\_GROUPCOMPONENT

So as an example, if your creating the component "com\_drwho", the JPATH\_GROUPCOMPONENT constant equals:

```
{web_root}/site/groups/{group_id}/components/com_drwho/
```

Note: You should be able to move the component to the main components folder and and have it work without any changes.

URL's built within a super group component will automatically have "/groups/{group\_cn}/" prepended to them. Please don't manually do that in your component or it will result in an error.

## Creating a Component

Creating components can always be done manually by creating the files in the correct location as described above. You can also utilize the Hubzero command line application. From the command line run the following command (in the web root):

```
{web_root}/cli/muse.php group scaffolding component --group={group_cn}  
-n=com_{component}
```

Simply replate {group\_cn} with your groups cname and {component} with the component. The component files will be automatically placed into the correct location, ready for you to modify and commit when ready.