

# pe\_mcollective

This module is specifically designed to manage the MCollective server authentication RSAkeys for use with Puppet Enterprise. Please see the documentation header for the `pe_mcollective` class for more information.

There are a small number of end-user facing pieces to this module; this is undesirable but is unavoidable if we want to enable users to use custom MCollective agent plugins in any capacity (we do). The "API" expectations, as outlined in [http://docs.puppetlabs.com/pe/3.0/orchestration\\_adding\\_actions.html](http://docs.puppetlabs.com/pe/3.0/orchestration_adding_actions.html), are:

- If a node's MCollective is being managed by PE, it is safe to use `include` on the `pe_mcollective` class. Nothing weird will happen.
  - Furthermore, doing so will reliably cause the `pe_mcollective::params` and `pe_mcollective::server::plugins` classes to be declared.
- The `pe_mcollective::server::plugins` class's `$plugin_basedir` variable will continue to be available and correct on all supported versions of all platforms.
- The `pe-mcollective` service's name is `pe-mcollective`. If a node's MCollective is being managed by PE, this service will always be present, will have the right name, and can be safely notified by other resources.
- The following variables from the `params` class will continue to be available and correct on all supported versions of all platforms:
  - `$pe_mcollective::params::mco_etc`
  - `$pe_mcollective::params::root_owner`
  - `$pe_mcollective::params::root_group`
  - `$pe_mcollective::params::root_mode`
- Implicitly, we're relying on the `plugins.d` directory already existing in MCollective's config directory. We think this is currently handled by the package/installer and not the module.

In return, users make the following promise:

- Users won't manage any `agent` or `data`-like subdirectories of the MCollective `libdir`. We will rely on the `pe_mcollective::server::plugins` class to do this, if it's necessary. (This is present because a pre-existing comment in the module implied that we may start managing more of them in the future.)

If these promises change, docs team needs a notification in pre-docs. Docs team will:

- Put a warning in the previous version's instructions.
- Change the upcoming version's instructions to match new behavior.
- Put a highly visible warning in the upgrade notes for the new version.
- Maintain that upgrade warning until the next MAJOR version, at which point we can roll it up into a "when upgrading from any prior version, make sure that..." type of note.

## Adding Tests

Please see the [rspec-puppet](#) project for information on writing tests. A basic test that validates the class is declared in the catalog is provided in the file `spec/classes/*_spec.rb`. `rspec-puppet` automatically uses the top level description as the name of a module to include in the catalog. Resources may be validated in the catalog using:

- `contain_class('myclass')`
- `contain_service('sshd')`
- `contain_file('/etc/puppet')`
- `contain_package('puppet')`
- And so forth for other Puppet resources.

## Running

To run all the `rspec` tests in the module, use the `spec` rake task.

```
# rake spec
```

## Note on identity

This version of the `pe_mcollective` module configure `mcollective` to use the puppet agent's node name as configured by the [node\\_name\\_value](#). Previously, the module used MCollective's default behavior of using the hostname of the node for the identity as it pertains to MCollective.

## Java Heap Size

The Java heap size may be tuned by setting the `activemq_heap_mb` class parameter. The value should be a string number representing the number of megabytes to allocate.

In order to support the Puppet Dashboard, this setting may be set as a Node parameter or as a Fact. If the class parameter is not set the value will be looked up in top scope. If it is not defined in top scope it will default to 512 MB.