Cluster Plugins

See also: Writing Registry Plugins

Some additional conventions are required when writing a Cluster Plugin.

1. The name of the Plugin must match the format `FooCluster`.
2. The Plugin must implement a model `FooCluster`, and a corresponding Controller. (These are in addition to the other models and controllers required for Plugins.)
    a. This Model should extend `ClusterInterface`, which defines some standard interfaces and provides some behind the scenes common functionality.
    b. The Controller should extend `SCController` ("Standard Cluster" Controller), which provides some common functionality.
    c. When a new Cluster Backend is instantiated, a skeletal row in the corresponding `co_foo_clusters` table will be created. There is no add operation or view required. The skeletal row will point to the parent Cluster Model.
    d. When a Cluster Backend is edited, the entry point to the Cluster Plugin will be `foo_cluster/foo_clusters/edit/#`. This will be called immediately after the Cluster Backend is first instantiated.
3. Note `Cluster` has a hasOne (ie: 1 to 1) relationship with `FooCluster`.
4. The table `cm_foo_clusters` should include a foreign key to `cm_clusters:id`.
    a. Other tables used by the plugin should reference `cm_foo_clusters:id`.
5. The Plugin must implement an `assign()` function, the signature of which is defined in `Model/ClusterInterface.php`. This function is called when automatic creation of an Account is created.
6. The Plugin must implement a `status()` function, the signature of which is defined in `Model/ClusterInterface.php`. This function returns an array, which currently holds one element: `comment`, a human readable description of the Cluster status for the specified CO Person.
7. ProvisionerBehavior will use `$cmPluginHasMany` (in `Model/FooCluster.php`) to determine which associated models to retrieve data for, in order to pass to Provisioning Plugins.
    a. Supported parent models are CoGroup and CoPerson. ie: In order to be provisioned, the associated model must have a foreign key `co_group_id` or `co_person_id`.
    b. Additionally, models associated with CoPerson must define the following fields:
        i. `status`: Only records with values `StatusEnum::Active` or `StatusEnum::GracePeriod` are provisioned.
        ii. `valid_from`: Records with values prior to "now", or a NULL value, are provisioned
        iii. `valid_through`: Records with values later than "now", or a NULL value, are provisioned
    c. For example, in the following configuration FooClusterAccounts will be available to Plugins:
        i. `public $cmPluginHasMany = array("CoPerson" => array("FooClusterAccount"));`

Note that, unlike Authenticator Plugins, there is currently no Registry-provided management functionality ("lock", "unlock", etc). Cluster Plugins must (currently) provide this functionality on their own.