Provisioner Plugins

See also: Writing Registry Plugins

Some additional conventions are required when writing a Provisioning Plugin.

⚠️ Provisioning Plugins are *Provision*ers*, but the parent model is *CoProvision*ing*. Be careful to reference the correct suffix.

1. The name of the Plugin should match the format `FooProvisioner`.
2. The Plugin should implement a model `CoFooProvisionerTarget`, and a corresponding Controller. (These are in addition to the other models and controllers required for Plugins.)
    a. This Model should extend `CoProvisionerPluginTarget`.
    b. The Controller should to extend `SPTController` ("Standard Provisioning Target" Controller), which provides some functionality common to most/all Provisioners (including parsing of `co_provisioning_target_id`).
    c. When a new Provisioning Target is created, a skeletal row in the corresponding `co_foo_provisioner_targets` table will be created. There is no `add` operation or view required. The skeletal row will point to the parent Provisioning Target.
    d. When a Provisioning Target is edited, the entry point to the Plugin will be `foo_provisioner/co_foo_provisioner_targets/edit/#/co:x`. This will be called immediately after the parent Provisioning Target is created.
3. Note `CoProvisioningTarget` has a `hasOne` (ie: 1 to 1) relationship with `CoFooProvisionerTarget`.
4. The table `co_foo_provisioner_target` should include a foreign key to `co_provisioning_target:id`.
    a. Other tables used by the plugin should reference `co_foo_provisioner_target:id`.
5. The Plugin must implement two functions, which are defined in `Model/CoProvisionerPluginTarget.php`. See that file for the function signatures, including what data is passed to the Plugin and what results are expected to be returned.
    a. `status()` to obtain the current provisioning status for a CO Person or CO Group. Note: there is a default implementation that may be sufficient for many provisioner plugins. Registry will automatically manage entries in cm_co_provisioning_exports and relay that information through the default `status()` call.
    b. `provision()` to execute provisioning for a CO Person, CO Group, CO Email List, or CO Service.

Provisioning and Delete Operations

When a delete operation passes through to a Provisioning Plugin, it will generally only be because a CoPerson (or CO Group) is deleted. (Deleting, say, a TelephoneNumber will show up as an update.) The plugin will be called *before* the delete has committed to the database. This is so that the underlying data is still available for the plugin to perform its work.

Note that cascading delete, described above, will clean up any relevant tables.

Obtaining Group Memberships

When a CO Group is provisioned, full group membership information is not provided with the provisioner, as groups can have sufficiently large numbers of members as to cause scalability issues. Instead, plugins requiring full group membership information should use the `CoGroupMember::findForCoGroup()` function and page over the results (or implement a similar method). This find will pull records for CO People who are not Active, so be sure to check for CO Person status if relevant to your plugin.

Group operations related to a single CO Person (eg: adding a CO Person to a CO Group) will include the relevant CO Person information in the provisioning data passed to the plugin.

Provisioner Assigned Identifiers

As of Registry v3.1.0, Identifiers may be associated with Provisioner Targets. This is useful if the plugin assigns or manages identifiers (for example, tracking the subject's identifier in the downstream system). Set `Identifier.co_provisioning_target_id` to the appropriate `co_provisioning_target:id`.

As of Registry v4.0.0, ProvisionerBehavior properly enforces data filtering based on status as documented. "No data provisioned" means that the provisioners may be invoked (depending on context), but will only be provided a CO Person ID and status. This minimal information is so that provisioners can deprovision an existing record that has been downgraded, if appropriate.