Provisioning From Registry

About Provisioning

Provisioning refers to action of using Registry data to create or remove access to applications and services. COmanage Registry defines a mechanism to extract Registry data for provisioning (via plugins), but there are multiple ways to provision applications from Registry:

1. Push Provisioning: Using Registry plugins, Registry notifies applications when relevant data has changed.
2. Pull Provisioning: Applications pull data from Registry on demand, either via the REST API or (less desirably) via database views.
3. Messaging: Upon change of relevant data, a message is issued to a Message Queue or Enterprise Service Bus, which is then responsible for distributing the message to the relevant downstream applications.

Push Provisioning

In order to enable Push Provisioning, one or more Provisioning Plugins must be installed. By default, Registry ships with various Provisioning Plugins, including for LDAP, Grouper, Mailman, and Salesforce. You can also write a custom plugin.

Adding a Provisioning Target

The first step to setting up provisioning is to define a *Provisioning Target*.

1. Login as a CO Administrator and select your CO.
2. Go to **Configuration > Provisioning Targets**.
3. Click **(+) Add Provisioning Target**.
4. Configure the new Provisioning Target.
    a. If the Plugin you select has additional configuration options (most do), they will appear after you click **Add.**
    b. Select the desired mode:
        i. **Automatic Mode**: The Plugin will be invoked automatically whenever COmanage Registry notices data suitable for provisioning has changed.
        ii. **Manual Mode**: The Plugin will only be invoked when a CO Admin explicitly does so (as described below).
        iii. **Enrollment Mode**: The Plugin will only be invoked once, at the conclusion of an Enrollment Flow. Available as of Registry v3. 2.0.
        iv. **Queue Mode**: The Plugin will not be immediately invoked, but instead a Job will be queued for asynchronous processing. See Queue Based Provisioning, below.
        v. **Queue on Error Mode**: The Plugin *will* be immediately invoked, but on error a Job will be queued for asynchronous reprocessing. See Queue Based Provisioning, below.
    c. As of Registry v1.0.3, Provisioning Targets can be ordered, so that a given provisioner can be guaranteed to run before another provisioner (for the same record). This applies only during Automatic Provisioning.
    d. As of Registry v2.0.0, a Provisioning Group can be specified. If provided, only CO People who are members of the CO Group (and only the CO Group itself, if the Provisioning Target also supports provisioning groups) will be provisioned using this provisioner. If a CO Person is subsequently removed from the group, their record will be deleted from the target (that is, the provisioning operation will automatically be converted to a delete action).

    ⚠ Keep in mind that when a CO Person is not in Active or Grace Period status, they are effectively removed from all Groups (with the exception of *All Members* Groups) for purposes of provisioning. So, for example, if a CO Person is expired, and the Provisioning Target is configured for the *I Like Ice Cream* Group (of which the person is a member), they will be completely removed from the Provisioning Target. Only *All Members* groups (at either the CO or COU level) can be used to maintain an expired record in the Provisioning Target (assuming the Target supports such behavior in the first place).

    e. As of Registry v3.2.0, an Organizational Identity Source can be associated with a Provisioning Target, via the **Skip If Associated With Org Identity Source** configuration option. If set, then a CO Person who has an Organizational Identity created from the specified Organizational identity Source will *not* be provisioned.

    ⚠ If used with a Provisioning Group, this setting takes precedence, and will prevent the CO Person from being provisioned.

5. Configure any Plugin specific options.

Manual Provisioning

Once one or more Provisioning Targets are defined, you can try them out manually by viewing any *active* CO Person record (**Organizations >** *YourCO* **>** **People > My Population**), clicking on **Provisioned Services**, and then clicking the **Provision** action.

Reprovisioning a Target

It is possible to reprovision all records for a given target, via **Organizations >** *YourCO* **> Configuration > Provisioning Targets** and then selecting the appropriate **Reprovision All** button. This effectively calls manual provisioning for each defined CO Person and CO Group (whether or not they are active). For large datasets, this operation may take a while.

Note that reprovisioning has no way of knowing how to clear entries from the provisioning target that are not known to COmanage Registry. A typical pattern for reprovisioning all records would be to first clear out the target entirely (eg: delete all records from the LDAP server) and then execute **Reprovisi on All**.

ⓘ As of Registry v3.3.0, **Reprovision All** will schedule a Registry Job using the Provisioner Job Plugin. (Prior to Registry v3.3.0, reprovisioning all ran synchronously upon request.) The Registry Job Shell must be set up to run queued jobs for reprovisioning to be processed.

Automatic Provisioning

Automatic Provisioning is triggered whenever data used for provisioning is changed. This data includes

- Address (attached to CO Person Role)
- CO Department
- CO Email List
- CO Group
- CO Group Membership
- CO Group Nesting
- CO Person Record
- CO Person Role Record
- CO Service
- CO Terms And Conditions Agreement
- Email Address (attached to CO Person)
- Identifier (attached to CO Person)
- Name (attached to CO Person)
- TelephoneNumber (attached to CO Person Role)
- URL (attached to CO Person)
- In general, Authenticators and Clusters, however it is up to each plugin to elect to do so

⚠ Which records are provisioned are determined by CO Person and Person Role Status.

> ⓘ **Note for Developers**
>
> Provisioning can be disabled on a per model/save basis by passing the option `"provision" => false` (v1.0.1 and later).

Queue Based Provisioning

As of Registry v4.0.0, queue based provisioning is supported in two ways: for all provisioning attempts, and on error only. A provisioning action is queued by scheduling a Registry Job using the Provisioner Job Plugin. For the queue to be processed, the Job Shell must be configured to run.

Only one Job for the combination of (Provisioning Target + Provisioning Subject + Provisioning Action) will be queued at any time. If a second Job is queued (for example, two actions causing provisioning happen quickly back to back), the second Job will be recorded as *Failed*, however the first Job will run and bring the target fully up to date. In the event of different actions (for example, an update followed by a delete), the actions will be executed sequentially.

Queue Based Provisioning, unlike Automatic Provisioning, is not necessarily immediate. For *Queue Mode*, the provisioning action will be queued for immediate execution. However, it will not be processed until the next run of Job Shell, which will depend on how often it is scheduled to run from cron and whether there are other jobs in the queue to be processed first. For *Queue On Error Mode*, the provisioning action *will* run immediately, but on failure will be subject to the *Retry Interval*.

The *Retry Interval* specifies how long to wait before trying to provision again after a failure (ie: the *first* time the action is queued in *Queue On Error Mode*, or the *second* time the action is queue in *Queue Mode*). The default is 900 seconds (15 minutes). Setting the interval to 0 will prevent the provisioning action from being tried again on a failure. A provisioning action stuck in a failure loop can be manually terminated by cancelling the currently queued job for the subject.

> ⓘ Queue Based Provisioning may be more efficient than Automatic Provisioning when a record changes multiple times in a short period (less than the queue processing interval). The multiple update events will effectively be collapsed into a single provisioning action.

Monitoring Push Provisioning

⚠ When provisioning is triggered automatically by an update, there is not currently a way to pass to the end user the results of the provisioning operation (other than manually clicking on the *Provisioned Services* link for the CO Person) (CO-582). If a provisioning plugin fails in such a situation, an error message will be syslog()d (at `LOG_ERR`). It is recommended that syslog be suitably configured and monitored to catch any errors with automatic provisioning. Alternately, as of Registry v4.0.0 use one of the Queue based operational modes.

Additionally, a Notification will be generated and sent to the CO Administrators.

Pull Provisioning

⚠ **Pull Provisioning Not Recommended**

Generally, pull provisioning from Registry is not recommended, as it ties applications tightly to the Registry implementation. Use of an intermediary such as LDAP is recommended.

Messaging

Support for message based provisioning is provided by the API Provisioning Plugin, available as of Registry v4.0.0.

Data Filtering

As of Registry v3.3.0, Data Filters can be used to modify data before it is passed to Provisioners.