Provisioning From Registry

Provisioning refers to the action of using Registry data to create or remove access to applications and services. COmanage Registry uses provisioning to provide the data that it stores to applications using one of three models, and in four different modes: Automatic, Manual, Enrollment-based, and Queue-based.

---

1. Provisioning Models

There are multiple models to provision applications from COmanage Registry:

1.1. Pull Provisioning

Applications pull data from COmanage Registry on demand, either via the REST API or (less desirable) via database views.

> ⚠ **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.

1.2. 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.

Support for message based provisioning is provided by the API Provisioning Plugin.

Registry v4.0.0 and higher

1.3. Push Provisioning

6. Configuring Provisioners in Registry

Provisioners are designed as a class of plugins. All plugins have basic settings that are are related to the plugin's Class. In addition, some plugins have plugin-specific settings to configure the specifics related to the plugin. To configure an Authenticator plugin in Registry you must:

1. Install and activate the Provisioner plugin. *(See the COmanage Registry Plugins page for additional details.)*
2. As the CO Administrator, navigate to the Provisioning Targets list for your CO by using the Configuration > Authenticators menu option.
3. Add a provisioning target by clicking on the [**Add Provisioning Target**] button on the right above the table.
4. This action will open a form to provide the Provisioning Plugin basic settings as listed below.

Field	Description
Description	The name that users will see when interacting with the provisioner. It should be descriptive of the provisioner.
Plugin	The plugin that will be used for this Provisioning Target
Status	The operating mode that should be used for this plugin. See the Provisioning Status section to the left for details. If the Status is set to "Disabled", the provisioner will not be available for execution.

Using Registry plugins, COmanage Registry notifies applications when relevant data has changed.

The remainder of the Technical Guide describes the details for configuring and using the **Push Provisioning Model** within COmanage Registry.

2. Terminology

There are multiple concepts with similar names. For clarity, here are their definitions:

- **Provisioning Plugin**: A COmanage Plugin, that implements the interfaces to provide data to a specific application (such as LDAP, Grouper, or a SQL-based database).
- **Provisioning Target**: An instantiated Provisioning Plugin. That is, a Provisioning Plugin with a specific configuration.

3. Provisioning Status (Operating Mode)

Provisioners may be operated by different modes:

1. Automatic Mode
2. Manual Mode
3. Enrollment Mode
4. Queue-based Provisioning
   a. Queue Mode
   b. Queue on Error Mode

This section describes how provisioning is affected based on the configured Status (Operating Mode)

3.1. Automatic Mode

The Plugin will be invoked automatically whenever COmanage Registry notices data suitable for provisioning has changed.

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

⚠️ The records that are provisioned are determined by CO Person and Person Role Status. The set of provisioned records may be limited further based on the Provisioning Target configuration. see the **Selective Provisioning Section** of this guide for more details.

ⓘ **Note for Developers**

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

Provisioning Group	Configuration that enables this Provisioning Target to apply only to a specific group. See the **Selective Provisioning section** of this guide for details.
Skip for Org Identity Source	Configuration that prevents this Provisioning Target from executing when a CO Person record has a specific type of Org Identity Source associated with it. See the **Selective Provisioning section** of this guide for details.
Order	Provisioning Targets can be ordered, so that a given provisioner can be guaranteed to run before another provisioner (for the same record). Run ordering only applies when Provisioning Targets are configured with an Automatic Operating Mode. *(Registry v1.0.3 and higher)*

5. When you have completed the form, click the [**ADD**] button to display a form to provide plugin-specific configurations (if any).

7. Using Provisioning

7.1. Manually Provisioning CO Person Records

All Provisioning Targets may be manually executed. For Targets that are configured with the Manual Mode status, manual execution is the only way that the Target will run. However, manual execution also can be helpful to test Provisioning Target configurations.

To manually execute a Provisioning Target:

1. View any *active* CO Person record. You may view the set of CO Person records for your CO by using the People > My Population menu option.
2. Click on the CO Person name or the [**Edit**] button to view the record and reveal the actions that may be taken on the record.
3. In the menu on the right, select the [**Provisioned Services**] link to display a list of active Provisioning Targets and the status of when they were last executed for this record.
4. Click on the [**Provision**] button to the right of the Provisioning Target to manually provision the CO Person record.

7.2. Reprovisioning a Target Application

It is possible to reprovision all records for a given target applications. This action effectively calls manual provisioning for each defined CO Person and CO Group (whether or not they are active).

To reprovision all records for a given target application:

1. Navigate to the Provisioning Targets list through the Configuration > Provisioning Target menu option.
2. Click on the [**Reprovision All**] button for the Provsioniong Target that you would like to reprovision.

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 **Reprovision All**.

Reprovisioning can take time

For large datasets, this reprovisioning operation may take a while.

ⓘ 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.

3.2. Manual Mode

The Plugin will only be invoked when a CO Administrator explicitly does so as described in the **Manually Provisioning CO Person Records Section** of this guide.

3.3. Enrollment Mode

Registry v3.2.0 and higher.

The Plugin will only be invoked once, at the conclusion of an Enrollment Flow.

3.4. Queue-based Provisioning

Registry v4.0.0 and higher

Queue based provisioning is supported in two ways: for all provisioning attempts (**Queue Mode**), and on error only (**Queue on Error Mode**). 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*.

> (i) 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.

When a Queue-based Operating Mode is selected, two additional configurations may be set on the Provisioning Target:

- **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.

- **Maximum Retry** — *(Registry v4.3.0 or higher)* Specifies how many times the job will try 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 3 times. Setting the value to 0 will prevent the provisioning action from being cancelled 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.

3.4.1. Queue Mode

8. 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.

The Plugin will not be immediately invoked, but instead a Job will be queued for asynchronous processing.

3.4.2. Queue on Error Mode

The Plugin will be immediately invoked, but on error a Job will be queued for asynchronous reprocessing.

4. Selective Provisioning

COmanage Registry allows for provisioning to happen only for CO Person records that meet specific criteria. The following filters may be set when configuring a Provisioning Target:

4.1. Provisioning Group

Registry v2.0.0 and higher

If configured, only CO People who are members of the configured CO Group will be provisioned using this provisioner. If a CO Person is subsequently removed from the group, their record will be deleted from the Provisioning Target application. That is, the provisioning operation will automatically be converted to a delete action.

Provisioning Group filters will also limit the CO Group that is provisioned to the one selected if the Provisioning Target also supports provisioning groups.

4.2. Org Identity Source

Registry v3.2.0 and higher

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. Registry Data Filtering

Registry v3.3.0 and higher

Data Filters are a plugin-based mechanism to modify or filter data in certain contexts. They can be used to modify data before it is passed to Provisioners. When a provisioning action is run, the relevant data is passed through any Data Filters attached to the Provisioning Target prior to the Provisioner Plugin receiving it.

See the Registry Data Filters guide for more details for configuration and use.

---

9. Extending with Plugins

The type and nature of provisioners used with the Registry can be extended through Plugins.

9.1. Provisioning Plugin Library

There are several plugins that are already available for your use:

10. See Also

10.1. Provisioner data model

The following database tables are associated with provisioners:

- cm_co_changelog_provisioner_targets — Per-CO Changelog provisioning target configurations

- API Provisioning Plugin — The API Provisioning Plugin provisions CO Person records to a RESTful or messaging endpoint.
- Changelog Provisioning Plugin — The Changelog Provisioning Plugin is a simple plugin that generates logfile entries on provisioning events. Entries are JSON encoded representations of CO Person and CO Group data.
- Crowd Provisioning Plugin — The Crowd Provisioning Plugin provisions CO Person and CO Group records to Atlassian Crowd.
- GitHub Provisioning Plugin — The GitHub Provisioning Plugin synchronizes Registry data with GitHub.
- Grouper Provisioning Plugin — The Grouper Provisioning Plugin provisions groups and memberships in groups to an Internet2 Grouper instance using the Grouper web services interface.
- Homedir Provisioning Plugin — The Homedir Provisioning Plugin is an experimental plugin that creates Unix home directories entries on provisioning events. It is not intended for use in a production environment. *(experimental)*
- Jira Provisioning Plugin — Registry v4.0.0 introduces the Jira Provisioning Plugin, which provisions CO Person and CO Group records to Atlassian Jira.
- LDAP Provisioning Plugin — The LDAP Provisioning Plugin is designed to provision Registry data into an LDAP server.
- Mailman Provisioning Plugin — The Mailman Provisioning Plugin manages Mailman3 mailing lists using Registry data. *(experimental)*
- MediaWiki Provisioning Plugin — The MediaWiki Provisioning Plugin provisions Registry data to a MediaWiki instance deployed with the OAuth extension. Since MediaWiki is not designed for group based authorization the plugin does not provision group information or memberships to MediaWiki. *(experimental)*
- MidPoint Provisioning Plugin — The MidPoint Provisioning Plugin provisions users to Evolveum midPoint using the midPoint REST API. *(experimental)*
- Salesforce Provisioning Plugin — The Salesforce Provisioning Plugin provisions Contacts to Salesforce via the Force.com REST API.
- SQL Provisioning Plugin — The SQL Provisioning Plugin provisions CO Person and CO Group records to a SQL database.

- cm_co_crowd_provisioner_targets — Per-CO Crowd provisioning target configurations
- cm_co_github_provisioner_targets — Per-CO GitHub provisioning target configurations
- cm_co_grouper_provisioner_groups — Per-CO per-Grouper target Grouper group map
- cm_co_grouper_provisioner_targets — Per-CO Grouper provisioning target configurations
- cm_co_homedir_provisioner_targets — Per-CO Home Directory provisioning target configurations
- cm_co_jira_provisioner_targets — Per-CO Jira provisioning target configurations
- cm_co_ldap_provisioner_attr_groupings — Per-CO per-LDAP target attribute grouping definitions
- cm_co_ldap_provisioner_attributes — Per-CO per-LDAP target attribute definitions
- cm_co_ldap_provisioner_dns — Per-CO per-LDAP target DN map
- cm_co_ldap_provisioner_targets — Per-CO LDAP provisioning target configurations
- cm_co_ldap_service_token_provisioner_targets — Per-CO Per-LDAP target service token provisioning target configurations
- cm_co_mailman_provisioner_targets — Per-CO Mailman provisioning target configurations
- cm_co_mid_point_provisioner_targets — Per-CO midPoint provisioning target configurations
- cm_co_provisioning_counts — Per-provisioning target job execution counts

9.2. Creating your own plugins

You can build your own authenticator plugin to extend Registry functionality. Please see the following documentation to get started:

- Writing Registry Plugins - general documentation for building plugins.
- Provisioner Plugins - additional documentation for building provisioning type plugins.