Registry Application Rules

Registry models are built on a basic CRUD model, however Registry implements additional logic in many places. This document provides a "contract" for these application (or "domain" or "business") rules that follows Semantic Versioning (beginning with Registry v5.0.0). ie: Rules documented here will not change in a backward-incompatible way between minor Registry versions.

General Model Rules (GMRs)

1. Once an entity is created within a CO, it cannot be moved to another CO.
2. Foreign keys from one entity to another cannot cross COs.
3. The Primary Link key for an object cannot be changed once set. (For example, an object that populates `person_id` cannot subsequently populate `external_identity_id` instead.)
4. Timestamps (regardless of which time and/or date components are significant) are stored in the database as UTC.
5. *Add*, *delete*, and *edit* actions on Provisionable Objects via StandardController or ApiV2Controller will trigger provisioning requests for the affected entity.

Plugin Application Rules (PARs)

Plugins may also define Application Rules, these are specified in the Plugin's documentation.

API User

1. API Users created in the COmanage CO (CO #1) have full access to the Registry API.
2. API Users created in other COs may be *privileged*, in which case they have full access to the Registry API for that CO, or *unprivileged*, in which case access most be contextually granted.
3. For namespacing purposes, API Users are named with a prefix consisting of the string `co_#.` (the letters `co`, an underscore, the numeric ID of the CO, and a dot). API usernames must be unique across the entire platform.
4. API Keys for API Users cannot be directly set, only generated.

Authentication Event

1. Authentication Events are tracked by authenticated identifiers, and therefore are not bound to any specific CO.
2. A CO Administrator may view all Authentication Events associated with any authenticated (login) Identifier associated with any Person within their CO, even if the Authentication Event was intended for access to another CO.
3. A Privileged CO API User may retrieve all Authentication Events associated with any authenticated (login) Identifier associated with any Person within its CO, even if the Authentication Event was intended for access to another CO.
4. Because Authentication Event records are not part of any CO, they are not deleted under circumstances such as the deletion of a CO.

CO

1. Deleting a CO will cause a hard delete of all CO related data.
2. The CO named `COmanage` cannot be rename, deleted, or suspended.
3. Two COs cannot share the same name.
4. Duplicating a CO will duplicate configuration related objects (such as COUs, Groups, and Enrollment Flows) but not data related objects (such as People and Group Memberships).
5. A CO cannot be deleted if it is in Active status.
6. When a new CO is created, the special groups associated with the CO will also be created.

7. If a CO is renamed, the special groups associated with the CO will also be renamed.

COU

1. A COU may not be deleted if it has any members (ie: People with a Person Role with the specified COU).
2. A COU may not be deleted if it has any children.
3. Two COUs within the same CO cannot share the same name.
4. When a COU is created, the special groups associated with the COU will also be created.
5. If a COU is renamed, the special groups associated with the COU will also be renamed.
6. If a COU is deleted, the special groups associated with the COU will also be deleted.

Enrollment Flow

1. An Enrollment Flow must have at least one Enrollment Flow Step defined in order to be run.

Enrollment Flow Step

1. Two Enrollment Flow Steps in the same Enrollment Flow cannot have the same order value.

Group

1. Two Groups within the same CO cannot share the same name.
2. A Group cannot be set to Suspended if it is nested into a Target Group or is a Target Group for a nesting.
3. A Group cannot be deleted if it is nested into a Target Group or is a Target Group for a nesting.
4. The *name*, *description*, and *status* of a Group of type *Owners* cannot be manually changed.
5. MVEAs cannot be attached to a Group of type *Owners*.
6. Groups of type *Owners* cannot be provisioned.
7. When a new Standard Group is created, if the Person who created it is not an Administrator, that Person will be set as the Group's initial Owner.
8. When a Group is deleted, its corresponding Owners Group is also deleted.
9. Standard Groups may not be named starting with the prefix `CO:`, which is reserved for System Groups.

Group Member

1. A Person cannot be given two manually created GroupMember records for the same Group. A Person can have more than one GroupMember record for the same Group via other mechanisms, such as Group Nestings, or External Identity Sources connected to Pipelines.
2. For Standard Groups, Group Memberships may be managed by any Group Member of the Group's corresponding Owners Group, and by any CO or COU Administrator.
3. For Owners Groups, Group Memberships may be managed by any Group Member of that Group, and by any CO or COU Administrator.

Group Nesting

1. Only Active Groups may be nested into a Target Group.
2. A Group may not be nested into itself.
3. A Group may not be nested into an Automatic Group.
4. A Group may not be nested into the same Target Group more than once, either directly or indirectly.
5. Group Nestings may not loop.

Identifier

1. The *login* flag may only be set on an Identifier if it is attached to a Person.
2. An Identifier must be (case sensitively) unique for the entity type within the CO.

Identifier Assignment

1. An IdentifierAssignment must apply to either an Identifier or an EmailAddress, but not both.
2. Availability checks for newly generated Identifiers and EmailAddresses are case insensitive.
3. EmailAddresses generated via Identifier Assignment are considered verified.

Job

1. A Job may not be registered if an existing Job with the same *plugin* and *parameters* is registered in either Queued or In Progress status, unless the Job Plugin supports.

Name

1. Exactly one Name for each Person must be designated as *Primary* at all times. The Primary Name cannot be deleted, another Name must be designated Primary first.

2. If a *display name* value is provided for any Name, it will be used within Registry whenever a full name representation of that Name is required.
3. If a full name is constructed from the various Name components, by default the components will be assembled as *given*, *middle*, *family*, *suffix*. However, for Names with a *language* set to *hu*, *ja*, *ko*, *za-Hans*, or *za-Hant*, the components will be assembled as *family*, *given*.
4. Each Person and External Identity must have at least one Name.

Person

1. When a Person is associated with a CO with any status other than *Archived*, the Person is added to the CO's *All Members* Group. If the Person is deleted from the COU, or the Person's status is set to *Archived*, the Person is removed from the CO's *All Members* Group.
2. When a Person is in *Active* or *Grace Period* status, the associated Person is added to the CO's *Active Members* Group. If the Person is set to any other status or is deleted from the , the Person is removed from the CO's *Active Members* Group.

Person Role

1. When a Person Role is associated with a COU with any status other than *Archived*, the associated Person is added to the COU's *All Members* Group. If the Person Role is disassociated from the COU, or the Person Role status is *Archived*, the associated Person is removed from the COU's *All Members* Group.
2. When a Person Role is in *Active* or *Grace Period* status and is associated with a COU, the associated Person is added to the COU's *Active Members* Group. If the Person Role is set to any other status or is disassociated from the COU, the associated Person is removed from the COU's *Active Members* Group.
3. A Person Role is considered valid (and provisionable) if (1) the Person Role is in *Active* or *Grace Period* status, (2) the valid from date is unspecified or in the past, and (3) the valid through date is unspecified or in the future.
4. A Person Role with a valid from date in the future and a status of *Active*, *Expired, or Grace Period* will be given a status of *Pending Activation*, unless the Person Role is frozen. A Person Role in *Pending Activation* status with a valid from date in the past will be given a status of *Active*.
5. A Person Role with a valid through date in the past and a status of *Active*, *Grace Period*, or *Pending Activation* will be given a status of *Expired*, unless the Person Role is frozen. A Person Role in *Expired* status with a valid from date in the future will be given a status of *Active*.
6. If both valid from and valid through dates are provided for a Person Role, the valid from date must be earlier than the valid through date.
7. When Person Role status changes, Person status will be recalculated, unless the Person status is *Locked*.

Pipeline

1. If a Pipeline creates a new Person, the first Name returned by the External Identity Source backend will be used as the initial Primary Name for the new Person.
2. Pipeline Person Matching ignores the existing Person status.

Plugin

1. Plugins must be uniquely named.
2. Plugins installed in the *core* directory cannot be disabled.
3. Plugins installed in the *available* directory are optional, and must be enabled prior to use.
4. Plugins installed in the *local* directory are deployment specific, and must be enabled prior to use.
5. If Plugins with the same name are installed in multiple locations, the resolution precedence is (1) *core*, (2) *available*, (3) *local*.
6. Plugin schemas, if defined, are applied to the database for active Plugins only.
7. If a Plugin is suspended, its associated database schema is *not* removed automatically.
8. A Plugin cannot be suspended or deleted if it is in use (referenced in a configuration object).
9. When a Plugin is instantiated, a skeletal record of the Entry Point Model is created, holding only a foreign key to the parent Pluggable Model. Validation and Application Rules are not executed when this record is created.
10. If a Pluggable Model is deleted, the Entry Point Model associated with the Pluggable Model is also deleted.
11. The Plugin Registry is refreshed upon loading of the Plugin Configuration page by the Platform Administrator.

Provisioning Target

1. After a new Provisioning Target is added, the associated Plugin cannot be changed.

Additional rules around Provisionable Objects and Eligibility are declared here.

Type

1. When a new CO is created, the default Types will be instantiated into the CO. Afterwards, available Types are only updated by administrator action.
2. A Type cannot be deleted once it has been used by at least one Registry object, even if that object is subsequently deleted.
3. A Suspended Type can not be assigned to new Registry objects, but existing objects already referencing it will not be changed.