# Authenticators

Authenticators are used to prove a CO Person's identity to an application or service. An Authenticator combined with an Identifier is a *credential*. Because Authenticators are collaboration issued, they are attached to the CO Person, not to Organizational Identities (as for external credentials).Although COmanage is built around the concept of external identity, there are a number of use cases where it makes sense for collaboration managed credentials to be used to access services, including

- Using SSH Keys (CO-1493) or Passwords to log in to unix based servers.
- Certificate access to grid computing resources.
- Multi factor authentication, using a collaboration issued second factor.

**On this page**

- 1. Terminology
- 2. How Authenticators Work
  - 2.1. Single vs Multiple Values
- 3. Authenticator Operations
  - 3.1. Notification On Update
  - 3.2. Linking to Management Page
- 4. Extending With Plugins
  - 4.1. Authenticator Plugin Library
  - 4.2. Creating your own plugins
- 5. See Also
  - 5.1. Authenticator data model
- 6. Configuring Authenticators in Registry
- 7. Using Authenticators To Log Into Registry

*Authenticators are available as of registry v3.1.0 (The SSH Key Authenticator plugin is available as of Registry v3.3.0. Prior to v3.3.0, SSH Key management is available via the CO Person canvas).*

## 1. Terminology

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

- **Authenticator Plugin**: A COmanage Plugin, that implements the interfaces to a specific authentication technology (such as Passwords or SSH Keys).
- **Authenticator Backend**: An instantiated Authenticator Plugin. That is, an Authenticator Plugin with a specific configuration.
- **Authenticator**: A specific instance of an authenticator attached to a CO Person. eg: A given person's password.

## 2. How Authenticators Work

Because Authenticators are collaboration issued, they are attached to the CO Person, not to Organizational Identities (as for external credentials). In general, Registry does not know how to validate Authenticators, they (or metadata about them) are simply stored in the Registry database. Authenticators are passed to the provisioning infrastructure, so that Provisioning Plugins may use the Authenticator information to populate downstream services. For example, the LDAP Provisioner Plugin may write a user password or SSH key attribute using Authenticator data.

### 2.1. Single vs Multiple Values

Authenticator Backends can support single or multiple values, as determined by the Authenticator Plugin. In general, whether an Authenticator Plugin supports multiple values depends on whether it makes sense for the CO Person to be able to manage multiple Authenticators of the same type for themselves.

For example, the Password Authenticator Plugin is single valued, meaning each instantiated backend may only have one password associated with it. (A given PasswordAuthenticator hasOne Password per CO Person.) If you want to support multiple passwords to be managed, you can instantiate multiple Backends. A CO Person cannot create a second password for themselves.

## 6. Configuring Authenticators in Registry

Authenticators 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 Authenticator plugin. *(See the COmanage Registry Plugins page for additional details.)*
2. As the CO Administrator, navigate to the Authenticator list for your CO by using the Configuration > Authenticators menu option.
   blocked URL
3. Add an authenticator by clicking on the [ **Add Authenticator** ] button on the right above the table.
   blocked URL
4. This action will open a form to provide the Authenticator Plugin basic settings as listed below.

| Field | Description |
|---|---|
| Description | The name that users will see when interacting with the authenticator. It should be descriptive of the authenticator. |
| Plugin | The plugin that will be used for this Authenticator. |
| Status | The Authenticator configuration can either be [ **Active** ] or [ **Suspended** ] |
| Change Message Template | *(optional)* When an authenticator is updated the associated CO Person will be sent a message using the specified message template. |

5. When you have completed the form, click the btn:[ADD] button to display a form to provide plugin-specific configurations (if any).
6. Active authenticators will be available in Registry where applicable.

On the other hand, the Certificate Authenticator Plugin is multi-valued, meaning each instantiated backend may support multiple certificates. (A given CertificateAuthenticator hasMany Certificate per CO Person.) This allows a CO Person to upload multiple certificates to attach to their operational record. (In general, there is no need to multiply instantiate an Authenticator Plugin that supports multiple values, although it is technically possible to do so.)

# 3. Authenticator Operations

Registry supports the following operations on Authenticators for a CO Person:

- **Manage**: Set or change the current Authenticator (for example, change a password). This operation may be performed by the CO Person (self service) or an administrator.
- **Lock**: Lock the Authenticator so it may not be changed or used. When locked, the Authenticator is not available to provisioners. This operation may only be performed by an administrator. For Authenticator Backends that support multiple values, locking applies to the entire Authenticator Backend (ie: all Authenticators for the CO Person, including the ability to add new ones).
- **Unlock**: Unlock the Authenticator so it may again be changed or used. If previously set, the original value will be maintained. This operation may only be performed by an administrator. For Authenticator Backends that support multiple values, unlocking applies to the entire Authenticator Backend.
- **Reset**: Clear the current Authenticator. This operation may only be performed by an administrator. Once reset, the CO Person may again manage the authenticator (if it is not locked). This operation is not supported for Authenticator Backends that support multiple values, although individual values maybe edited or deleted.

Upon any operation, the provisioning infrastructure is executed so that the Provisioners may perform any necessary actions. Authenticators may be manually reprovisioned by the usual manual reprovisioning process.

As of Registry v3.3.0, Authenticators may be collected as part of an Enrollment Flow.

## 3.1. Notification On Update

As of Registry v4.0.0, notifications may be sent when an Authenticator is updated. To enable this capability, first define a Message Template with a context of *Authenticator*. Then, edit the desired Authenticator configuration. Set the *Change Message Template* to the newly created template.

⚠ Custom Authenticator plugins may need to manually trigger this notification. If writing a custom plugin, see the Authenticator Plugins documentation for more information.

## 3.2. Linking to Management Page

As of Registry v4.0.0, URLs of the form /registry/password_authenticator/passwords/manage/authenticatorid:X (ie: that do not specifically include a CO Person ID) will automatically redirect to the current CO Person's Authenticator management page (or require login if there is no current session). This is useful to offer a generic "Manage My Credential" link from a user portal or documentation source.

# 4. Extending With Plugins

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

## 4.1. Authenticator Plugin Library

# 7. Using Authenticators To Log Into Registry

Because Authenticators are attached to the CO Person, not to Organizational Identities, and because Registry does not know how to validate Authenticators, they cannot be directly used to log into Registry. The following additional steps are necessary:

1. Set up an authentication service (eg: Shibboleth, CAS, or mod_authnz_ldap) that allows for web based authentication using the Authenticators as provisioned to a suitable target (such as LDAP), and configure Apache to use this authentication service for Registry.
2. For each CO Person, create an Org Identity (or use an existing one) with an attached Identifier flagged for login.

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

- Certificate Authenticator Plugin — The Certificate Authenticator plugin manages information about X.509 Certificates for CO People. *(experimental)*
- Password Authenticator Plugin — The Password Authenticator plugin manages passwords for CO People. *(experimental)*
- Privacy IDEA Authenticator Plugin — The Privacy IDEA Authenticator plugin provides an interface for tokens managed by a privacyIDEA authentication server. These tokens can be used to implement Multi-Factor Authentication (though instructions for doing so are beyond the scope of this document). *(experimental)*
- SSH Key Authenticator Plugin — The SSH Key Authenticator plugin manages SSH Public Keys for CO People.

## 4.2. 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.
- Authenticator Plugins - additional documentation for building authenticator type plugins.

# 5. See Also

## 5.1. Authenticator data model

The following database tables are associated with authenticators:

- cm_authenticator_statuses — Authenticator Statuses
- cm_authenticators — Authenticators