

Extending the Registry Data Model

- [Extending Attributes](#)
 - [Ad Hoc Attributes](#)
 - [Extended Attributes](#)
 - [Implementation Notes](#)
 - [DIY \(Do It Yourself\)](#)
- [Extended Types](#)
 - [eduPersonAffiliation and Extended Affiliations](#)

Extending Attributes

The standard [Registry Data Model](#) is unlikely to cover all attributes required for a given CO. There are three ways of extending the data model to add additional attributes.

Ad Hoc Attributes

Available as of Registry v3.3.0, Ad Hoc Attributes are simple key/value pairs attached to CO Person Roles or Organizational Identities. Ad Hoc Attributes are available via the [REST API](#), are supported by [Organizational Identity Source Plugins](#) and [Pipelines](#), and are made available to [Provisioner Plugins](#).

Ad Hoc Attributes are not currently validated in any way, do not support Enrollment Flows ([CO-1478](#)), and are not available to attach to the CO Person ([CO-469](#)).



Because Ad Hoc Attributes are neither namespaced nor typed, they should not be used as a replacement for solutions leveraging properly designed structured data. Ad Hoc Attributes are most suitable for simple, local data extensions where the cost of structured data is not warranted.

Extended Attributes



Extended Attributes are deprecated as of Registry v3.3.0. Extended Attributes will be converted to Ad Hoc Attributes in a future release ([CO-1478](#)).

The Registry supports simple *extended attributes* that can be attached to CO Person Role records on a per-CO basis. These extended attributes can be managed entirely via the Registry UI (select "Extended Attributes" from the CO menu) or the [REST API](#), however the following limitations currently apply:

- Extended attributes only permit one value per CO Person Role. For example, if you wanted to add an attribute for a favorite ice cream flavor, each person can have only one favorite ice cream flavor. (As a workaround, you could define two similar fields: favorite1 and favorite2.)
- The only data types currently supported are:
 - INTEGER
 - TIMESTAMP
 - VARCHAR(32)
- Once defined, an attribute's type cannot be changed.
- Complex relationships, such as foreign keys, are not supported.

Implementation Notes

The information in this section is primarily of use for database administrators who need to know what is going on under the hood.

When an extended attribute is added to a CO, a record of it is added to the [cm_co_extended_attributes](#) table. However, this record serves primarily as an index or reference. The actual extended attribute values attached to a given CO Person Role are stored in dynamically managed tables.

The first extended attribute to be added to a CO will trigger the creation of a new table of the pattern *cm_co#_person_extended_attributes*, where # is the numeric ID of the relevant CO. Each extended attribute that is defined gets a column in the table, and as values are set for individual CO Person Roles, each individual Person Role receives a row in this table. If an extended attribute is set to index, the appropriate index is automatically created on this dynamic table.

Deleting an extended attribute causes the associated column to be dropped from the table. If all extended attributes are deleted from a CO, then the dynamic table is dropped as well.

DIY (Do It Yourself)

More sophisticated requirements can be met by implementing native models, views, and controllers in the CakePHP framework. The best way to do that is to write a [Plugin](#).

Extended Types

Extended types allow custom types to be defined for a given attribute. This doesn't directly change the Registry Data Model, but allows flexibility in how some attributes are used. For example, several email address types are defined by default (delivery, forwarding, official, and personal). However, it may be desirable to add additional types (such as pager or deprecated). This is done via Extended Types.

Not all attributes support Extended Types. Defining custom types for supported attributes is done by selecting "Extended Types" from the CO menu.

The following attributes support Extended Types:

- Address Type
- CO Department Type (as of Registry v3.3.0)
- CO Person Role Affiliation
- Email Address Type
- Identifier Type
- Name Type
- Organization Type (as of Registry v4.0.0)
- Telephone Number Type

If an Extended Type is in use (by specifying an attribute of the type, or by referencing the type in an Enrollment Flow), it cannot be deleted or suspended.

The default types for any Extended Type can be restored by selecting the appropriate Extended Type and clicking *Add/Restore Default Types*. Any existing non-default types will remain.

 As of Registry v4.1.0, Org Identities support Extended Types. Within a CO, CO Person and Org Identity attributes share the same set of available types. Org Identity Affiliation will use the same options as CO Person Role Affiliation.

eduPersonAffiliation and Extended Affiliations

[eduPersonAffiliation](#) is a controlled vocabulary with limited pre-defined values that may not be suitable for all internal purposes. As such, the CO Person Role Affiliation is an *Extended Type*, and may be customized to add additional values (such as undergraduate and graduate).

Nevertheless, it may be useful to express these additional values in terms of eduPersonAffiliations. For example, if the VO attributes are to be expressed in a federated environment, using the controlled vocabulary (student instead of undergraduate or graduate) may be desirable.

When defining Extended Affiliations, the custom affiliations can be mapped to [eduPersonAffiliations](#) by selecting the appropriate value from the popup while adding or editing the custom affiliation. If a mapping is left blank, the eduPersonAffiliation will automatically map to an exact match (ie: student = student), otherwise a blank mapping is ignored.

This mapping is used by the [LDAP Provisioning Plugin](#), which uses the mapping to populate the eduPersonAffiliation attribute, and the [SQL Provisioning Plugin](#).