

Upgrading Registry to v0.9.2

- [Extended Types](#)
- [Fix Incorrect Foreign Key](#)
- [Automated Processing \(via cron\)](#)
- [Add CMP Enrollment Attribute Types](#)

Extended Types

Registry v0.9.2 introduces Extended Types for additional attributes, and changes how Extended Types work for Identifiers (which already supported them). This is a fairly disruptive change and requires the following steps to be run:

1. Make sure the database schema has updated successfully. Several fields are made wider. In particular, Postgres databases may need to be manually updated due to [CO-165](#).

```
SQL> ALTER TABLE cm_addresses ALTER COLUMN type TYPE VARCHAR(32);
SQL> ALTER TABLE cm_co_person_roles ALTER COLUMN affiliation TYPE VARCHAR(32);
SQL> ALTER TABLE cm_email_addresses ALTER COLUMN type TYPE VARCHAR(32);
SQL> ALTER TABLE cm_names ALTER COLUMN type TYPE VARCHAR(32);
SQL> ALTER TABLE cm_org_identities ALTER COLUMN affiliation TYPE VARCHAR(32);
SQL> ALTER TABLE cm_telephone_numbers ALTER COLUMN type TYPE VARCHAR(32);
```

2. Convert any existing Identifier extended types:

```
SQL> UPDATE cm_co_extended_types SET attribute='Identifier.type' WHERE attribute='Identifier';
```

3. Instantiate the new default types. This must be done **for each CO** you have defined, including the CManage CO.
 - a. Select *Collaborations >> CO >> Configuration >> Extended Types*
 - b. For each attribute in the *For Attribute* popup, select the attribute from the popup and then click *Add/Restore Default Types*.
4. Update existing type values for the new types (note multiple formats given for the same commands, so if you run the POSTGRES commands you don't need to run the SQL commands):

```
-- Address.type
POSTGRES> update cm_addresses set type = v.newstr
from (values
  ('H', 'home'),
  ('O', 'office'),
  ('P', 'postal')
) as v(oldstr, newstr)
where v.oldstr = cm_addresses.type;

SQL> update cm_addresses set type='home' where type='H';
SQL> update cm_addresses set type='office' where type='O';
SQL> update cm_addresses set type='postal' where type='P';

-- CoPersonRole.affiliation
POSTGRES> update cm_co_person_roles set affiliation = v.newstr
from (values
  ('AF', 'affiliate'),
  ('AL', 'alum'),
  ('E', 'employee'),
  ('F', 'faculty'),
  ('SA', 'staff'),
  ('SU', 'student'),
  ('M', 'member'),
  ('L', 'librarywalkin')
) as v(oldstr, newstr)
where v.oldstr = cm_co_person_roles.affiliation;

SQL> update cm_co_person_roles set affiliation='affiliate' where affiliation='AF';
SQL> update cm_co_person_roles set affiliation='alum' where affiliation='AL';
SQL> update cm_co_person_roles set affiliation='employee' where affiliation='E';
SQL> update cm_co_person_roles set affiliation='faculty' where affiliation='F';
SQL> update cm_co_person_roles set affiliation='staff' where affiliation='SA';
```

```

SQL> update cm_co_person_roles set affiliation='student' where affiliation='SU';
SQL> update cm_co_person_roles set affiliation='member' where affiliation='M';
SQL> update cm_co_person_roles set affiliation='librarywalkin' where affiliation='L';

-- EmailAddress.type
POSTGRES> update cm_email_addresses set type = v.newstr
from (values
      ('D', 'delivery'),
      ('F', 'forwarding'),
      ('O', 'official'),
      ('P', 'personal')
     ) as v(oldstr, newstr)
where v.oldstr = cm_email_addresses.type;

SQL> update cm_email_addresses set type='delivery' where type='D';
SQL> update cm_email_addresses set type='forwarding' where type='F';
SQL> update cm_email_addresses set type='official' where type='O';
SQL> update cm_email_addresses set type='personal' where type='P';

-- Name.type
POSTGRES> update cm_names set type = v.newstr
from (values
      ('L', 'alternate'),
      ('A', 'author'),
      ('F', 'fka'),
      ('O', 'official'),
      ('P', 'preferred')
     ) as v(oldstr, newstr)
where v.oldstr = cm_names.type;

SQL> update cm_names set type='alternate' where type='L';
SQL> update cm_names set type='author' where type='A';
SQL> update cm_names set type='fka' where type='F';
SQL> update cm_names set type='official' where type='O';
SQL> update cm_names set type='preferred' where type='P';

-- OrgIdentity.affiliation
POSTGRES> update cm_org_identities set affiliation = v.newstr
from (values
      ('AF', 'affiliate'),
      ('AL', 'alum'),
      ('E', 'employee'),
      ('F', 'faculty'),
      ('SA', 'staff'),
      ('SU', 'student'),
      ('M', 'member'),
      ('L', 'librarywalkin')
     ) as v(oldstr, newstr)
where v.oldstr = cm_org_identities.affiliation;

SQL> update cm_org_identities set affiliation='affiliate' where affiliation='AF';
SQL> update cm_org_identities set affiliation='alum' where affiliation='AL';
SQL> update cm_org_identities set affiliation='employee' where affiliation='E';
SQL> update cm_org_identities set affiliation='faculty' where affiliation='F';
SQL> update cm_org_identities set affiliation='staff' where affiliation='SA';
SQL> update cm_org_identities set affiliation='student' where affiliation='SU';
SQL> update cm_org_identities set affiliation='member' where affiliation='M';
SQL> update cm_org_identities set affiliation='librarywalkin' where affiliation='L';

-- TelephoneNumber.type
POSTGRES> update cm_telephone_numbers set type = v.newstr
from (values
      ('F', 'fax'),
      ('H', 'home'),
      ('M', 'mobile'),
      ('O', 'office')
     ) as v(oldstr, newstr)
where v.oldstr = cm_telephone_numbers.type;

SQL> update cm_telephone_numbers set type='fax' where type='F';
SQL> update cm_telephone_numbers set type='home' where type='H';

```

```
SQL> update cm_telephone_numbers set type='mobile' where type='M';
SQL> update cm_telephone_numbers set type='office' where type='O';
```

5. Update existing enrollment flow attributes for the new types:

```
POSTGRES> update cm_co_enrollment_attributes set attribute = v.newstr
from (values
 ('i:address:H', 'i:address:home'),
 ('i:address:O', 'i:address:office'),
 ('i:address:P', 'i:address:postal'),
 ('i:address:R', 'i:address:forwarding'),
 ('i:email_address:D', 'i:email_address:delivery'),
 ('i:email_address:F', 'i:email_address:forwarding'),
 ('i:email_address:O', 'i:email_address:official'),
 ('i:email_address:P', 'i:email_address:personal'),
 ('i:name:L', 'i:name:alternate'),
 ('i:name:A', 'i:name:author'),
 ('i:name:F', 'i:name:fka'),
 ('i:name:O', 'i:name:official'),
 ('i:name:P', 'i:name:preferred'),
 ('i:telephone_number:F', 'i:telephone_number:fax'),
 ('i:telephone_number:H', 'i:telephone_number:home'),
 ('i:telephone_number:M', 'i:telephone_number:mobile'),
 ('i:telephone_number:O', 'i:telephone_number:office'),
 ('m:address:H', 'm:address:home'),
 ('m:address:O', 'm:address:office'),
 ('m:address:P', 'm:address:postal'),
 ('m:address:R', 'm:address:forwarding'),
 ('m:telephone_number:F', 'm:telephone_number:fax'),
 ('m:telephone_number:H', 'm:telephone_number:home'),
 ('m:telephone_number:M', 'm:telephone_number:mobile'),
 ('m:telephone_number:O', 'm:telephone_number:office'),
 ('p:email_address:D', 'p:email_address:delivery'),
 ('p:email_address:F', 'p:email_address:forwarding'),
 ('p:email_address:O', 'p:email_address:official'),
 ('p:email_address:P', 'p:email_address:personal'),
 ('p:name:L', 'p:name:alternate'),
 ('p:name:A', 'p:name:author'),
 ('p:name:F', 'p:name:fka'),
 ('p:name:O', 'p:name:official'),
 ('p:name:P', 'p:name:preferred'))
) as v(oldstr, newstr)
where v.oldstr = cm_co_enrollment_attributes.attribute;

SQL> update cm_co_enrollment_attributes set attribute='i:address:home' where attribute='i:address:H';
SQL> update cm_co_enrollment_attributes set attribute='i:address:office' where attribute='i:address:O';
SQL> update cm_co_enrollment_attributes set attribute='i:address:postal' where attribute='i:address:P';
SQL> update cm_co_enrollment_attributes set attribute='i:address:forwarding' where attribute='i:address:R';
SQL> update cm_co_enrollment_attributes set attribute='i:email_address:delivery' where attribute='i:email_address:D';
SQL> update cm_co_enrollment_attributes set attribute='i:email_address:forwarding' where attribute='i:email_address:F';
SQL> update cm_co_enrollment_attributes set attribute='i:email_address:official' where attribute='i:email_address:O';
SQL> update cm_co_enrollment_attributes set attribute='i:email_address:personal' where attribute='i:email_address:P';
SQL> update cm_co_enrollment_attributes set attribute='i:name:alternate' where attribute='i:name:L';
SQL> update cm_co_enrollment_attributes set attribute='i:name:author' where attribute='i:name:A';
SQL> update cm_co_enrollment_attributes set attribute='i:name:fka' where attribute='i:name:F';
SQL> update cm_co_enrollment_attributes set attribute='i:name:official' where attribute='i:name:O';
SQL> update cm_co_enrollment_attributes set attribute='i:name:preferred' where attribute='i:name:P';
SQL> update cm_co_enrollment_attributes set attribute='i:telephone_number:fax' where attribute='i:telephone_number:F';
SQL> update cm_co_enrollment_attributes set attribute='i:telephone_number:home' where attribute='i:telephone_number:H';
SQL> update cm_co_enrollment_attributes set attribute='i:telephone_number:mobile' where attribute='i:telephone_number:M';
SQL> update cm_co_enrollment_attributes set attribute='i:telephone_number:office' where attribute='i:telephone_number:O';
```

```

telephone_number:0';
SQL> update cm_co_enrollment_attributes set attribute='m:address:home' where attribute='m:address:H';
SQL> update cm_co_enrollment_attributes set attribute='m:address:office' where attribute='m:address:O';
SQL> update cm_co_enrollment_attributes set attribute='m:address:postal' where attribute='m:address:P';
SQL> update cm_co_enrollment_attributes set attribute='m:address:forwarding' where attribute='m:address:
R';
SQL> update cm_co_enrollment_attributes set attribute='m:telephone_number:fax' where attribute='m:
telephone_number:F';
SQL> update cm_co_enrollment_attributes set attribute='m:telephone_number:home' where attribute='m:
telephone_number:H';
SQL> update cm_co_enrollment_attributes set attribute='m:telephone_number:mobile' where attribute='m:
telephone_number:M';
SQL> update cm_co_enrollment_attributes set attribute='m:telephone_number:office' where attribute='m:
telephone_number:O';
SQL> update cm_co_enrollment_attributes set attribute='p:email_address:delivery' where attribute='p:
email_address:D';
SQL> update cm_co_enrollment_attributes set attribute='p:email_address:forwarding' where attribute='p:
email_address:F';
SQL> update cm_co_enrollment_attributes set attribute='p:email_address:official' where attribute='p:
email_address:O';
SQL> update cm_co_enrollment_attributes set attribute='p:email_address:personal' where attribute='p:
email_address:P';
SQL> update cm_co_enrollment_attributes set attribute='p:name:alternate' where attribute='p:name:L';
SQL> update cm_co_enrollment_attributes set attribute='p:name:author' where attribute='p:name:A';
SQL> update cm_co_enrollment_attributes set attribute='p:name:fka' where attribute='p:name:F';
SQL> update cm_co_enrollment_attributes set attribute='p:name:official' where attribute='p:name:O';
SQL> update cm_co_enrollment_attributes set attribute='p:name:preferred' where attribute='p:name:P';

```

6. If any enrollment flows have an *affiliation* attribute with a default value, the default value must be re-selected manually (edit the appropriate enrollment flow attribute), even if it appears that the default value is correct.
7. For any defined LDAP Provisioners, decide whether or not to provision the employeeType attribute. This must be done regardless of the decision. Edit the appropriate provisioning target(s) and scroll to the list of attributes. Check or uncheck employeeType, as appropriate. Click Save, whether or not any changes were made.
8. For any defined LDAP Provisioners, reselect the types to export for the following attributes. This must be done for any enabled attribute, even if it appears the correct value is already selected:
 - telephoneNumber
 - facsimileTelephoneNumber
 - address
 - mail
 - mobile
9. If CO Self Service Permissions are defined, they may need to be updated for the new types:

```

-- Address
POSTGRES> update cm_co_self_service_permissions set type = v.newstr
from (values
      ('H', 'home'),
      ('O', 'office'),
      ('P', 'postal')
     ) as v(oldstr, newstr)
where v.oldstr = cm_co_self_service_permissions.type
and model = 'Address';

SQL> update cm_co_self_service_permissions set type='home' where type='H' and model = 'Address';
SQL> update cm_co_self_service_permissions set type='office' where type='O' and model = 'Address';
SQL> update cm_co_self_service_permissions set type='postal' where type='P' and model = 'Address';

-- EmailAddress
POSTGRES> update cm_co_self_service_permissions set type = v.newstr
from (values
      ('D', 'delivery'),
      ('F', 'forwarding'),
      ('O', 'official'),
      ('P', 'personal')
     ) as v(oldstr, newstr)
where v.oldstr = cm_co_self_service_permissions.type
and model = 'EmailAddress';

SQL> update cm_co_self_service_permissions set type='delivery' where type='D' and model =
'EmailAddress';
SQL> update cm_co_self_service_permissions set type='forwarding' where type='F' and model =
'EmailAddress';
SQL> update cm_co_self_service_permissions set type='official' where type='O' and model =
'EmailAddress';
SQL> update cm_co_self_service_permissions set type='personal' where type='P' and model =
'EmailAddress';

-- Name
POSTGRES> update cm_co_self_service_permissions set type = v.newstr
from (values
      ('L', 'alternate'),
      ('A', 'author'),
      ('F', 'fka'),
      ('O', 'official'),
      ('P', 'preferred')
     ) as v(oldstr, newstr)
where v.oldstr = cm_co_self_service_permissions.type
and model = 'Name';

SQL> update cm_co_self_service_permissions set type='alternate' where type='L' and model = 'Name';
SQL> update cm_co_self_service_permissions set type='author' where type='A' and model = 'Name';
SQL> update cm_co_self_service_permissions set type='fka' where type='F' and model = 'Name';
SQL> update cm_co_self_service_permissions set type='official' where type='O' and model = 'Name';
SQL> update cm_co_self_service_permissions set type='preferred' where type='P' and model = 'Name';

-- TelephoneNumber
POSTGRES> update cm_co_self_service_permissions set type = v.newstr
from (values
      ('F', 'fax'),
      ('H', 'home'),
      ('M', 'mobile'),
      ('O', 'office')
     ) as v(oldstr, newstr)
where v.oldstr = cm_co_self_service_permissions.type
and model = 'TelephoneNumber';

SQL> update cm_co_self_service_permissions set type='fax' where type='F' and model = 'TelephoneNumber';
SQL> update cm_co_self_service_permissions set type='home' where type='H' and model = 'TelephoneNumber';
SQL> update cm_co_self_service_permissions set type='mobile' where type='M' and model =
'TelephoneNumber';
SQL> update cm_co_self_service_permissions set type='office' where type='O' and model =
'TelephoneNumber';

```

10. If any Identifier Assignments are defined with type mail, update email_type:

```
POSTGRES> update cm_co_identifier_assignments set email_type = v.newstr
from (values
      ('D', 'delivery'),
      ('F', 'forwarding'),
      ('O', 'official'),
      ('P', 'personal')
     ) as v(oldstr, newstr)
where v.oldstr = cm_co_identifier_assignments.email_type
and identifier_type = 'mail';

SQL> update cm_co_identifier_assignments set email_type='delivery' where email_type='D' and
identifier_type = 'mail';
SQL> update cm_co_identifier_assignments set email_type='forwarding' where email_type='F' and
identifier_type = 'mail';
SQL> update cm_co_identifier_assignments set email_type='official' where email_type='O' and
identifier_type = 'mail';
SQL> update cm_co_identifier_assignments set email_type='personal' where email_type='P' and
identifier_type = 'mail';
```

11. cm_co_petition_attributes should be updated, although this is not strictly necessary. This table holds *archived* petition attributes – ie their values at time of petition creation. Updating these values will allow them to render correctly via the UI, but no other functionality is otherwise impacted.

```
-- CoPetitionAttributes.affiliation
POSTGRES> update cm_co_petition_attributes set value = v.newstr
from (values
      ('AF', 'affiliate'),
      ('AL', 'alum'),
      ('E', 'employee'),
      ('F', 'faculty'),
      ('SA', 'staff'),
      ('SU', 'student'),
      ('M', 'member'),
      ('L', 'librarywalkin')
     ) as v(oldstr, newstr)
where v.oldstr = cm_co_petition_attributes.value and
cm_co_petition_attributes.attribute='affiliation';

SQL> update cm_co_petition_attributes set value='affiliate' where value='AF' and
cm_co_petition_attributes.attribute='affiliation';
SQL> update cm_co_petition_attributes set value='alum' where value='AL' and
cm_co_petition_attributes.attribute='affiliation';
SQL> update cm_co_petition_attributes set value='employee' where value='E' and
cm_co_petition_attributes.attribute='affiliation';
SQL> update cm_co_petition_attributes set value='faculty' where value='F' and
cm_co_petition_attributes.attribute='affiliation';
SQL> update cm_co_petition_attributes set value='staff' where value='SA' and
cm_co_petition_attributes.attribute='affiliation';
SQL> update cm_co_petition_attributes set value='student' where value='SU' and
cm_co_petition_attributes.attribute='affiliation';
SQL> update cm_co_petition_attributes set value='member' where value='M' and
cm_co_petition_attributes.attribute='affiliation';
SQL> update cm_co_petition_attributes set value='librarywalkin' where value='L' and
cm_co_petition_attributes.attribute='affiliation';
```

Fix Incorrect Foreign Key

Prior versions of the database schema incorrectly created a foreign key from cm_co_terms_and_conditions:cou_id to cm_co_groups:id instead of cm_cous:id. Running the database schema update *should* fix this, but it is best to verify this manually.

To manually fix the foreign key (Postgres):

```
postgres=> alter table cm_co_terms_and_conditions drop constraint cm_co_terms_and_conditions_cou_id_fkey;
postgres=> alter table cm_co_terms_and_conditions add constraint cm_co_terms_and_conditions_cou_id_fkey foreign
key (cou_id) references cm_cous(id);
```

Automated Processing (via cron)

Registry v0.9.2 also introduces cron based automatic processing. [Install the cron command](#) to enable this processing.

Add CMP Enrollment Attribute Types

If CMP Enrollment Attributes have been defined (typically for [populating organizational identity attributes from authoritative sources](#)), types must be added for many of these attributes:

```
SQL> update cm_cmp_enrollment_attributes set type='official' where attribute like 'names:%' or attribute like
'email_addresses:%';
SQL> update cm_cmp_enrollment_attributes set type='office' where attribute like 'addresses:%' or attribute like
'telephone_numbers:%';
```

Also, ePPN is no longer a CMP Enrollment Attribute:

```
SQL> delete from cm_cmp_enrollment_attributes where type='eppn' and attribute = 'identifiers:identifier';
```