Creating custom entry forms

- Overview
- An example
- How is it created?

Overview

Custom entry forms allow VIVO to transcend the general-purpose, utilitarian editing scheme of Vitro. Without custom entry forms, VIVO users must edit each RDF triple individually. With a custom entry form, users can edit a complex data structure on a single page.
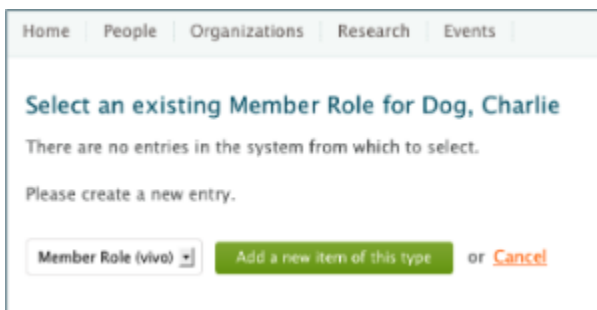
VIVO is distributed with a dozens of custom entry form generators. You may want to modify these form generators, or add more of your own.

An example

Say you wish to establish that a particular person is a member of a particular academic department. This relationship can be expressed as a member role. See Membership Model

But what if the academic department doesn't exist in VIVO yet? You will want to create that department, and assign a name to it. You may also want to record the member role in that department, when their membership began, and when it ended (if it is not ongoing).

Without a custom entry form, you would need to record each piece of data individually.
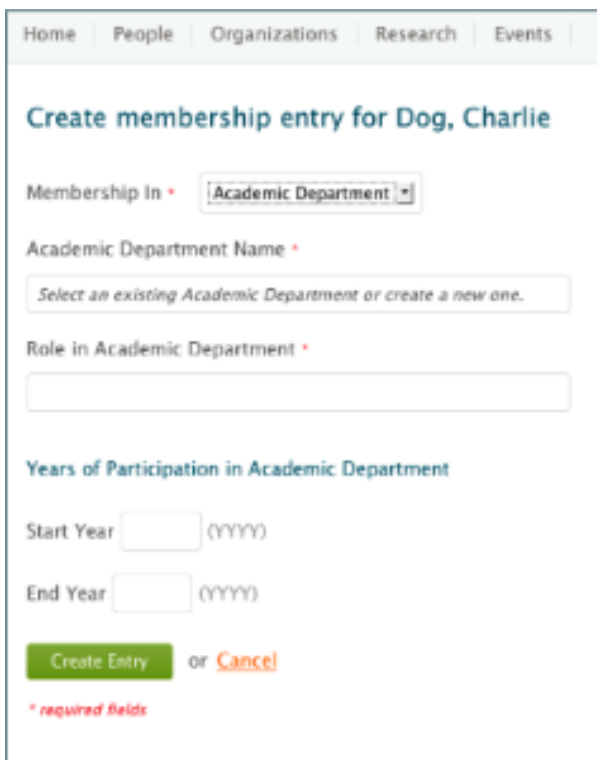
VIVO includes a custom form generator for this relationship. The custom entry form looks like this:

How is it created?

The creation of custom entry forms is an arcane and eldritch art, for which little documentation is available.

Each form requires a Java class known as a `EditConfigurationGenerator`. The generator describes the data structure being created, lists the SPARQL queries used, and includes a reference to the Freemarker template that will render the form.

You can start by examining the existing generators in this directory
`[VIVO]/src/edu/cornell/mannlib/vitro/webapp/edit/n3editing/configuration/generators`

and the Freemarker templates found here
`[VIVO]/productMods/templates/freemarker/edit/forms`

- Note: The directory structure has changed in version 1.9+.
  [VIVO]/src/... is now [VIVO]/api/src/main/java/...
  [VIVO]/productMods/... is now [VIVO]/webapp/src/main/webapp/...

There is also a short page of technical description called Implementing custom forms using N3 editing.