

Create class named DemoProjectRegistry

stereotype: folder

tagged values:

```
archetype_name = Project Registry
typeDescription = A project registry
content_icon = navigation_icon.gif
other values: author, email, license
```

attributes:

```
description:text      required=0,searchable=1
```

run AGX

edit DemoProjectRegistry.py and add:

```
_at_rename_after_creation = True
```

Create project registry content

Create class named DemoProject

tagged values:

```
archetype_name = Project
typeDescription = A project.
```

add composition from ProjectRegistry

attributes

```
description:text
```

```
note:text
```

```
problemIssue:text    description=The problem issue being addressed by this project.,
```

schemata=project details

```
projectObjective:text description=A brief description of the project's objectives.,
```

schemata=project details, label=Objective

run AGX

edit DemoProject.py to add _at_rename_after_creation = True

create project content

Edit note attribute

```
widget = RichTextWidget
```

```
widget:label, widget:description
```

run AGX, reload product

AGX will not prevent you from making mistakes

correct description widget to be RichWidget

run AGX

Create class named DemoOrganization

tagged values

archetype_name, description

attributes

organizationType:string label=Type,description=Select the classification which best describes this organization,enforceVocabulary=1,vocabulary=python:['Academic Research Organization (ARO)','Clinical Research Organization (CRO)','Device Company','Pharmaceutical Company',]
isPathsPartner:boolean label:Organization is a PATHs partner, description=
url:string label:Web site,description=The organization's primary Web address,validators=isURL

add composition from Project Registry

run AGX

edit DemoProject.py to add _at_rename_after_creation = True

Create class named DemoParticipant

tagged values

archetype_name, description

attributes

email:string label:E-mail Address,description,validations=isEmail
professionalTitle:lines label=ProfessionalTitle, description=Select the person's title,
widget:SelectionWidget,enforceVocabulary=1,vocabulary:type=ATVocabularyManager,vocabulary:na
me=DemoProfessionalTitle

add composition from Project Registry

Run AGX

edit DemoParticipant.py to add _at_rename_after_creation = True

edit AppConfig.py to add dependency on ATVocabularyManager

add Participant, show vocabulary, then cancel

goto control panel and setup vocabulary

add several terms

add participant

add association btw participant and organization. Rename the endpoints. Add label to ending point.

Run AGX

update schema

edit the participant

add association btw participant and project. Rename the endpoints and add label, description to end point.

Since we want to use ReferenceWidget, add to class

tagged values

```
imports = from Products.ATReferenceBrowserWidget.ATReferenceBrowserWidget import  
ReferenceBrowserWidget
```

update association endpoint

tagged values

```
relationship = project_participant  
multiValued = python:1  
widget = ReferenceBrowserWidget
```

run AGX

create a back reference for project participants

edit project class

add attribute participants:Backreference

tagged values

```
relationship = project_participant, widget:visible=python:'{edit:invisible}',multiValued=1  
edit AppConfig to add dependency for ATBackRef
```

run AGX

edit project registry

add method summary, stereotype=view

naming the method view would override base_view, same type of thing for edit and contents

run AGX

show workflow using DCRIWorkflow