Grouper Loader classlist example from Penn

Wiki Home	Grouper Release Announcements	Grouper Guides	Grouper Deployment Guide	Community Contributions	Internal Developer Resources

Main Grouper Loader page

Summary

University of Pennsylvania implemented class-lists with the Grouper Loader which includes lists of students, instructors, assistants (to instructors), guests, and all members. The students, instructors, and assistants are fed from our student system, so they should be include/exclude. The guests are an ad hoc group. The "all" members should include the groups: students, instructors, assistants, guests. There can be cross listings (multiple names for a course), and the cross listing groups should just contain the primary course groups (5 groups: all, students, instructors, guests, assistants). Security should be setup on all courses such that there is a global readers group, a global updaters group, and in each course the instructors and assistants should be able to read and update the appropriate groups.   Note you need Grouper 1.4 built after 10/18/2009, or 1.5+ for all of this to work properly.

Naming

Our naming convention is similar to name groups like this (this is just one course of many):

penn:community:student:**course**   -   root of all courses
penn:community:student:**loader**   - holds groups with loader rules
penn:community:student:**security**   - holds security groups (e.g. readers or updaters of all courses)

penn:community:student:course:2009C:EG:CIS:120:203:**all**

- holds all members for term 2009C, engineering, computer science, course 120, section 203
- includes the students, instructors, assistants, and guests

penn:community:student:course:2009C:EG:CIS:120:203:**assistants**
 - overall assistants to instructors (system of record, add includes, remove excludes)
penn:community:student:course:2009C:EG:CIS:120:203:**assistants excludes**
 -assistants to instructors excludes list (remove from system of record)
penn:community:student:course:2009C:EG:CIS:120:203:**assistants includes**
 - assistants to instructors includes list
penn:community:student:course:2009C:EG:CIS:120:203:**assistants_systemOfRecord**
 - assistants to instructors system or record from student system
penn:community:student:course:2009C:EG:CIS:120:203:**guests**
 - ad hoc guests group to course
penn:community:student:course:2009C:EG:CIS:120:203:**instructors**
 - overall instructors group (system of record, add includes, subtract excludes)
penn:community:student:course:2009C:EG:CIS:120:203:**instructors excludes**
 - instructors excludes, will block members from system of record
penn:community:student:course:2009C:EG:CIS:120:203:**instructors includes**
 - instructors includes, to add members to the system of record
penn:community:student:course:2009C:EG:CIS:120:203:**instructors_systemOfRecord**
 - instructors system of record, form student system
penn:community:student:course:2009C:EG:CIS:120:203:**students**
 - overall students list, made up of system of record, add includes, subtract excludes
penn:community:student:course:2009C:EG:CIS:120:203:**students excludes**
 - students excludes removes students from the system of record
penn:community:student:course:2009C:EG:CIS:120:203:**students includes**
 - students includes add members to the system of record
penn:community:student:course:2009C:EG:CIS:120:203:**students_systemOfRecord**
 - students system of record is fed from student system

Security design

There are two high level global security groups for system or admins:

penn:community:student:security:courseReaders   (can read all course membership lists)
penn:community:student:security:courseUpdaters   (can update all course membership lists)

The instructors and assistants can read all groups in the course (listed above), and all cross-listed courses to that course.

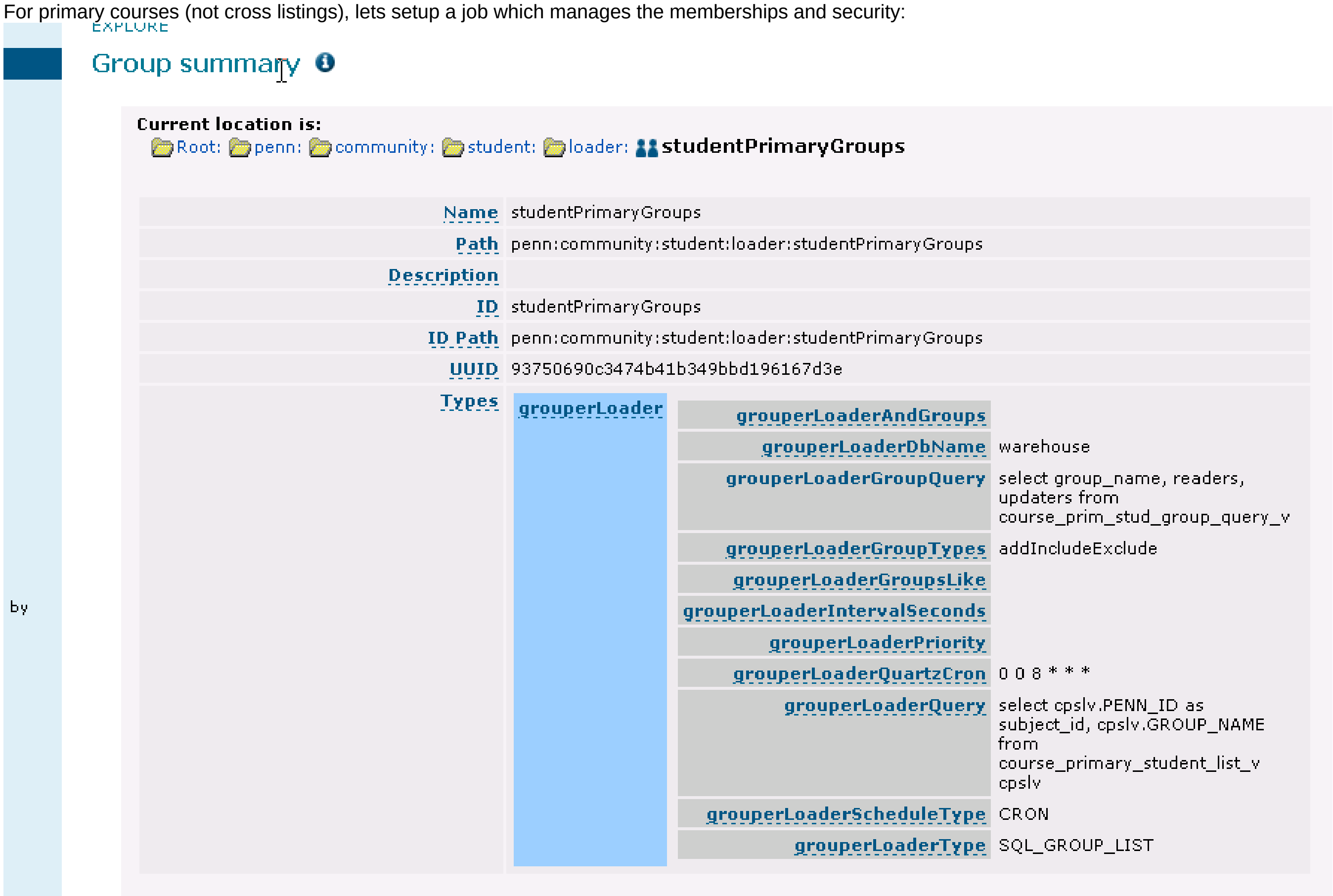
They can update:

- guests
- instructors includes
- instructors excludes
- students includes
- students excludes
- assistants includes
- assistants excludes

Note, no cross listed courses are editable since they mirror the primary course

Primary course students list

For primary courses (not cross listings), lets setup a job which manages the memberships and security:



EXPLORE

Group summary

Current location is:
Root: penn: community: student: loader: **studentPrimaryGroups**

Name	studentPrimaryGroups
Path	penn:community:student:loader:studentPrimaryGroups
Description	
ID	studentPrimaryGroups
ID Path	penn:community:student:loader:studentPrimaryGroups
UUID	93750690c3474b41b349bbd196167d3e
Types	grouperLoader

grouperLoaderAndGroups	
grouperLoaderDbName	warehouse
grouperLoaderGroupQuery	select group_name, readers, updaters from course_prim_stud_group_query_v
grouperLoaderGroupTypes	addIncludeExclude
grouperLoaderGroupsLike	
grouperLoaderIntervalSeconds	
grouperLoaderPriority	
grouperLoaderQuartzCron	0 0 8 * * *
grouperLoaderQuery	select cpslv.PENN_ID as subject_id, cpslv.GROUP_NAME from course_primary_student_list_v cpslv
grouperLoaderScheduleType	CRON
grouperLoaderType	SQL_GROUP_LIST

This shows:

- The data is coming from our warehouse (configured in grouper-loader.properties)
- There is a grouper group type applied to the group, which is addIncludeExclude. This automatically creates the includes, excludes, and overall groups
- The job runs once per day at 8am
- It is a SQL_GROUP_LIST which manages many groups at once
- The membership query: (penn_id and group_name one), returns data like this:
  ```
  12345678    penn:community:student:course:2009C:EG:BE:099:001:students_systemOfRecord
  12345679    penn:community:student:course:2009C:EG:BE:100:001:students_systemOfRecord
  12345677    penn:community:student:course:2009C:EG:BE:100:001:students_systemOfRecord
  12345676    penn:community:student:course:2009C:EG:BE:100:001:students_systemOfRecord
  12345675    penn:community:student:course:2009C:EG:BE:101:001:students_systemOfRecord
  12345678    penn:community:student:course:2009C:EG:BE:101:001:students_systemOfRecord
  ```
- The groups query, controls empty groups, names groups, and sets the security. It returns data that looks like this:

GROUP_NAME	READERS	UPDATERS
penn:community:student:course:2009C:EG:BE:099:001:students_systemOfRecord	penn:community:student:security:courseReaders, penn:community:student:course:2009C:EG:BE:099:001:assistants, penn:community:student:course:2009C:EG:BE:099:001:instructors	penn:community:student:security:courseUpdaters, penn:community:student:course:2009C:EG:BE:099:001:assistants, penn:community:student:course:2009C:EG:BE:099:001:instructors
penn:community:student:course:2009C:EG:BE:100:001:students_systemOfRecord	penn:community:student:security:courseReaders, penn:community:student:course:2009C:EG:BE:100:001:assistants, penn:community:student:course:2009C:EG:BE:100:001:instructors	penn:community:student:security:courseUpdaters, penn:community:student:course:2009C:EG:BE:100:001:assistants, penn:community:student:course:2009C:EG:BE:100:001:instructors
penn:community:student:course:2009C:EG:BE:100:201:students_systemOfRecord	penn:community:student:security:courseReaders, penn:community:student:course:2009C:EG:BE:100:201:assistants, penn:community:student:course:2009C:EG:BE:100:201:instructors	penn:community:student:security:courseUpdaters, penn:community:student:course:2009C:EG:BE:100:201:assistants, penn:community:student:course:2009C:EG:BE:100:201:instructors

- Note that the queries in the loader job are built on views, it is important that you do this so you can see when upgrades make things not compile, and so the SQL can be changed without editing the job, and since there is a character limit for attribute values
- First I create a stem name view, which includes if a course is primary or not (cross listed). Note this also restricts which terms to select from

```
CREATE OR REPLACE VIEW COURSE_GROUP_STEM_NAME_V
(GROUP_NAME_STEM, TERM, SECTION_ID, XLIST_PRIMARY, PRIMARY_COURSE)
AS select 'penn:community:student:course:' || trim(cs.term) || ':' || trim(cs.section_school)
            || ':' || trim(cs.subject_area)
           || ':' || trim(cs.course_number) || ':' || trim(cs.section_number)
            as group_name_stem,
           cs.term, cs.section_id, cs.XLIST_PRIMARY, decode(cs.SECTION_ID, cs.XLIST_PRIMARY, 'T', 'F') as
primary_course
           from course_section cs, course_section cs_xlist_primary, course_term_v ctv ...
```

- Then I built on that to create a course group name which has all the names of the groups (for easy reuse)

```
CREATE OR REPLACE VIEW COURSE_GROUP_NAME_V
(GROUP_NAME_STEM, STUDENTS, STUDENTS_SYSTEMOFRECORD, STUDENTS_INCLUDES, STUDENTS_EXCLUDES,
 INSTRUCTORS, INSTRUCTORS_SYSTEMOFRECORD, INSTRUCTORS_INCLUDES, INSTRUCTORS_EXCLUDES, ASSISTANTS,
 ASSISTANTS_SYSTEMOFRECORD, ASSISTANTS_INCLUDES, ASSISTANTS_EXCLUDES, ALLGROUP, GUESTS,
 SECTION_ID, TERM, XLIST_PRIMARY, PRIMARY_COURSE)
AS
select
cgsnv.GROUP_NAME_STEM,
cgsnv.GROUP_NAME_STEM || ':students' as students,
cgsnv.GROUP_NAME_STEM || ':students_systemOfRecord' as students_systemOfRecord,
cgsnv.GROUP_NAME_STEM || ':students_includes' as students_includes,
cgsnv.GROUP_NAME_STEM || ':students_excludes' as students_excludes,
cgsnv.GROUP_NAME_STEM || ':instructors' as instructors,
cgsnv.GROUP_NAME_STEM || ':instructors_systemOfRecord' as instructors_systemOfRecord,
cgsnv.GROUP_NAME_STEM || ':instructors_includes' as instructors_includes,
cgsnv.GROUP_NAME_STEM || ':instructors_excludes' as instructors_excludes,
cgsnv.GROUP_NAME_STEM || ':assistants' as assistants,
cgsnv.GROUP_NAME_STEM || ':assistants_systemOfRecord' as assistants_systemOfRecord,
cgsnv.GROUP_NAME_STEM || ':assistants_includes' as assistants_includes,
cgsnv.GROUP_NAME_STEM || ':assistants_excludes' as assistants_excludes,
cgsnv.GROUP_NAME_STEM || ':all' as allGroup,
cgsnv.GROUP_NAME_STEM || ':guests' as guests,
cgsnv.SECTION_ID,
cgsnv.TERM,
cgsnv.XLIST_PRIMARY,
cgsnv.PRIMARY_COURSE
from COURSE_GROUP_STEM_NAME_V cgsnv
```

- Then we can build course list view (note the members of cross listed courses are includes in the primary course)

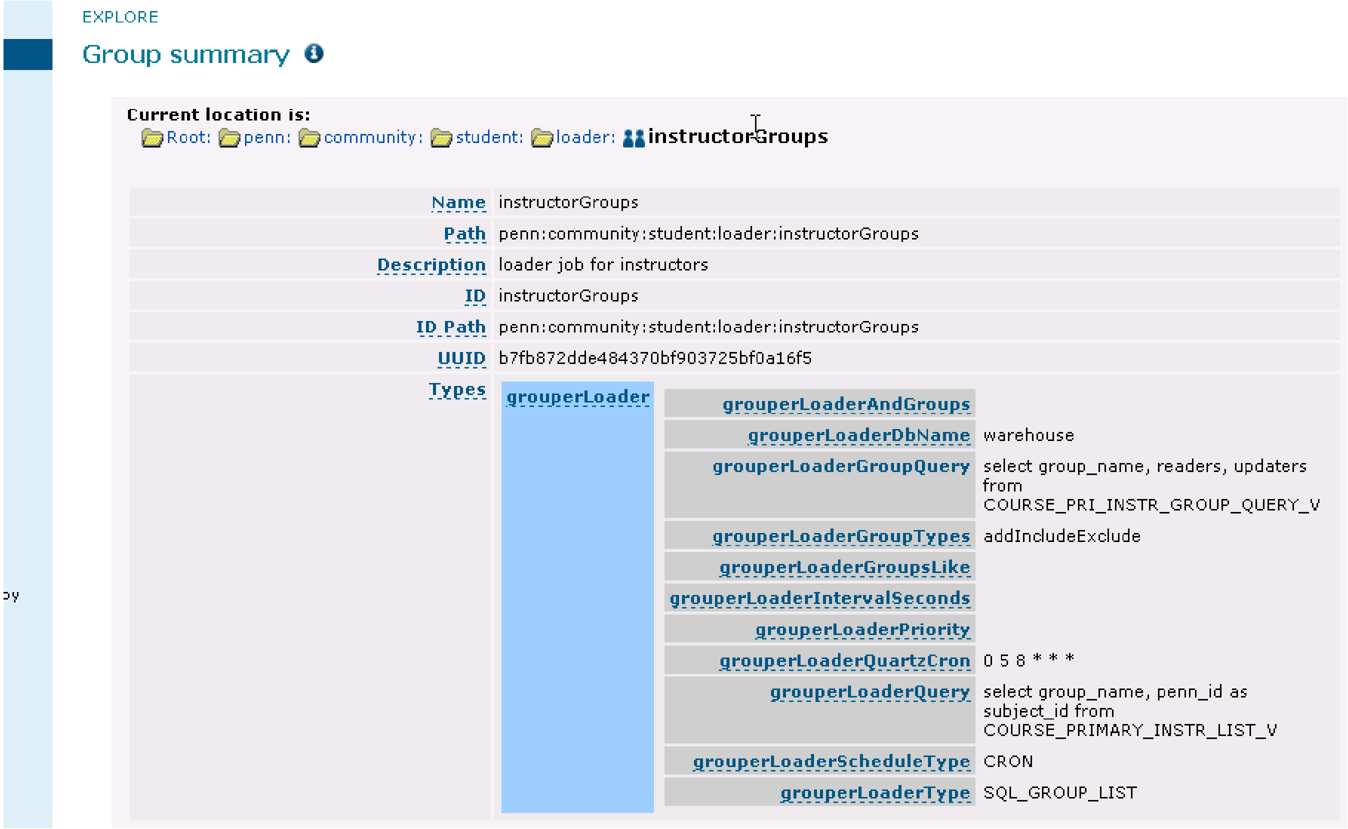
```
CREATE OR REPLACE VIEW COURSE_PRIMARY_STUDENT_LIST_V
(GROUP_NAME, PENN_ID, TERM, SECTION_ID)
AS
SELECT
  distinct cgnv.STUDENTS_SYSTEMOFRECORD as group_name,
  eav.PENN_ID,
  eav.TERM,
  cs.XLIST_PRIMARY section_id
FROM
  DWADMIN.ENROLLMENT_ALL_V  eav,
  DWADMIN.COURSE_SECTION  cs,
  course_group_name_v cgnv
WHERE
  eav.SECTION_ID=cs.section_id and eav.TERM=cs.TERM
  and cgnv.SECTION_ID = cs.XLIST_PRIMARY
  and cgnv.TERM = cs.term
  and cgnv.PRIMARY_COURSE = 'T'
```

- Finally we make the primary students group view for the students list which includes the security aspects

```
CREATE OR REPLACE VIEW COURSE_PRIM_STUD_GROUP_QUERY_V
(GROUP_NAME, READERS, UPDATERS, SECTION_ID, TERM)
AS
select cgnv.STUDENTS_SYSTEMOFRECORD as group_name,
'penn:community:student:security:courseReaders, ' || cgnv.ASSISTANTS
|| ',' || cgnv.INSTRUCTORS as readers,
'penn:community:student:security:courseUpdaters, ' || cgnv.ASSISTANTS
|| ',' || cgnv.INSTRUCTORS as updaters, cgnv.SECTION_ID, cgnv.TERM
from course_group_name_v cgnv where cgnv.PRIMARY_COURSE = 'T'
```

Primary course instructors list

Note: this is called instructors since it is more of a role than a title, so it is not "Professors". Here is the loader screenshot:



Current location is:
Root: penn: community: student: loader: instructorGroups

Name	instructorGroups
Path	penn:community:student:loader:instructorGroups
Description	loader job for instructors
ID	instructorGroups
ID Path	penn:community:student:loader:instructorGroups
UUID	b7fb872dde484370bf903725bf0a16f5
Types	grouperLoader

grouperLoader	
grouperLoaderAndGroups	
grouperLoaderDbName	warehouse
grouperLoaderGroupQuery	select group_name, readers, updaters from COURSE_PRI_INSTR_GROUP_QUERY_V
grouperLoaderGroupTypes	addIncludeExclude
grouperLoaderGroupsLike	
grouperLoaderIntervalSeconds	
grouperLoaderPriority	
grouperLoaderQuartzCron	0 5 8 * * *
grouperLoaderQuery	select group_name, penn_id as subject_id from COURSE_PRIMARY_INSTR_LIST_V
grouperLoaderScheduleType	CRON
grouperLoaderType	SQL_GROUP_LIST

- This is a query which runs from our warehouse connection (configured in grouper-loader.properties), add include/exclude type to the groups for include exclude lists, runs at 8:05 in the morning
- The membership query builds on a view which gives instructors or admins for a course (or any of its cross listings)

```
select cpiav.INSTRUCTOR_PENN_ID penn_id,
cpiav.instructors_GROUP_NAME as group_name
from COURSE_PRIMARY_INSTR_ASST_V cpiav
where cpiav.INSTRUCTOR_LOAD <> 0
```

- This returns data like this:

PENN_ID	GROUP_NAME
12345678	penn:community:student:course:2009C:EG:BE:200:204:instructors_systemOfRecord
12345678	penn:community:student:course:2009C:EG:MGMT:586:501: instructors_systemOfRecord
12345677	penn:community:student:course:2009C:EG:MSE:990:001:instructors_systemOfRecord
12345676	penn:community:student:course:2009C:EG:BE:497:001:instructors_systemOfRecord

- The groups view gives the name of the group and the security rules

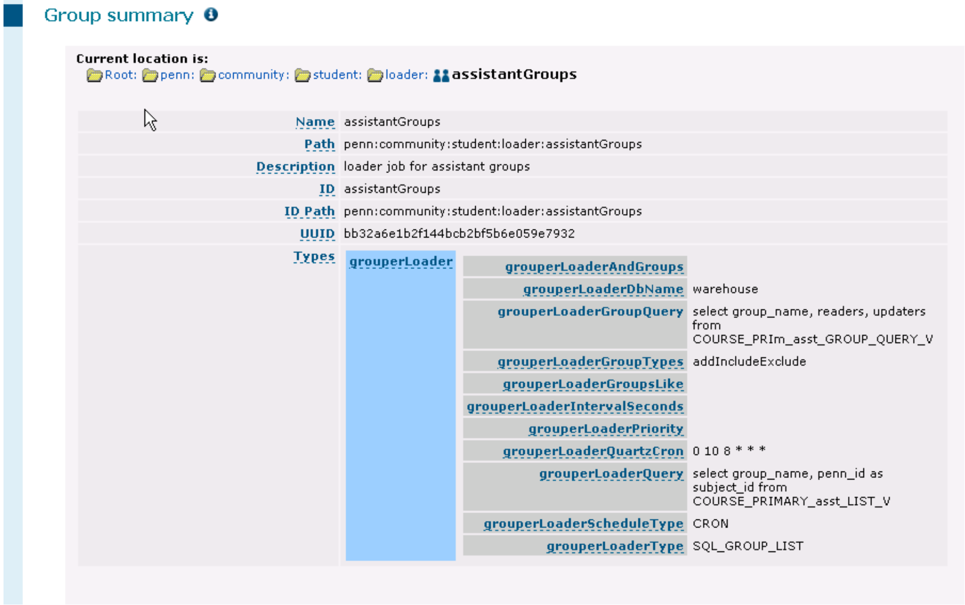
```
CREATE OR REPLACE VIEW COURSE_PRI_INSTR_GROUP_QUERY_V
(GROUP_NAME, READERS, UPDATERS, SECTION_ID, TERM)
AS
select cgnv.instructors_SYSTEMOFRECORD as group_name,
'penn:community:student:security:courseReaders, ' || cgnv.ASSISTANTS
|| ',' || cgnv.INSTRUCTORS as readers,
'penn:community:student:security:courseUpdaters, ' || cgnv.ASSISTANTS
|| ',' || cgnv.INSTRUCTORS as updaters, cgnv.SECTION_ID, cgnv.TERM
from course_group_name_v cgnv
```

- This returns data that looks like this:

GROUP_NAME	READERS	UPDATERS	SECTION_ID	TERM
penn:community:student:course:2009C:AS:ANTH:258:401:instructors_systemOfRecord	penn:community:student:security:courseReaders, penn:community:student:course:2009C:AS:ANTH:258:401:assistants,penn:community:student:course:2009C:AS:ANTH:258:401:instructors	penn:community:student:security:courseUpdaters, penn:community:student:course:2009C:AS:ANTH:258:401:assistants,penn:community:student:course:2009C:AS:ANTH:258:401:instructors	ANTH258401	2009C
penn:community:student:course:2009C:FA:ARCH:727:401:instructors_systemOfRecord	penn:community:student:security:courseReaders, penn:community:student:course:2009C:FA:ARCH:727:401:assistants,penn:community:student:course:2009C:FA:ARCH:727:401:instructors	penn:community:student:security:courseUpdaters, penn:community:student:course:2009C:FA:ARCH:727:401:assistants,penn:community:student:course:2009C:FA:ARCH:727:401:instructors	ARCH727401	2009C

Primary course assistants list

- Note, this is not called "teaching assistants" since this group is more of a role than a title



Group summary

Current location is:
Root: penn: community: student: loader: assistantGroups

Name	assistantGroups	
Path	penn:community:student:loader:assistantGroups	
Description	loader job for assistant groups	
ID	assistantGroups	
ID Path	penn:community:student:loader:assistantGroups	
UUID	bb32a6e1b2f144bcb2bf5b6e059e7932	
Types	grouperLoader	
	grouperLoaderAndGroups	
	grouperLoaderDbName	warehouse
	grouperLoaderGroupQuery	select group_name, readers, updaters from COURSE_PRIm_asst_GROUP_QUERY_V
	grouperLoaderGroupTypes	addIncludeExclude
	grouperLoaderGroupsLike	
	grouperLoaderIntervalSeconds	
	grouperLoaderPriority	
	grouperLoaderQuartzCron	0 10 8 * * *
	grouperLoaderQuery	select group_name, penn_id as subject_id from COURSE_PRIMARY_asst_LIST_V
	grouperLoaderScheduleType	CRON
	grouperLoaderType	SQL_GROUP_LIST

- Here is the loader screenshot
- This runs from the data warehouse (configured in grouper-loader.properties), at 8:10 in the morning, and includeExclude is added for the include and exclude lists
- Here is a view of the members:

```
select cpiav.INSTRUCTOR_PENN_ID penn_id,
cpiav.ASSISTANTS_GROUP_NAME as group_name
from COURSE_PRIMARY_INSTR_ASST_V cpiav
where cpiav.INSTRUCTOR_LOAD = 0
```

- Here is a sample of the data returned:

PENN_ID	GROUP_NAME
38430684	penn:community:student:course:2009C:EG:MEAM:210:203:assistants_systemOfRecord
45535464	penn:community:student:course:2009C:EG:CIS:120:203:assistants_systemOfRecord
24636453	penn:community:student:course:2009C:EG:ESE:112:001:assistants_systemOfRecord

- Here is the view of the group query:

```
CREATE OR REPLACE VIEW COURSE_PRIM_ASST_GROUP_QUERY_V
(GROUP_NAME, READERS, UPDATERS, SECTION_ID, TERM)
AS
select cgnv.assistants_SYSTEMOFRECORD as group_name,
'penn:community:student:security:courseReaders, ' || cgnv.ASSISTANTS
|| ',' || cgnv.INSTRUCTORS as readers,
'penn:community:student:security:courseUpdaters, ' || cgnv.ASSISTANTS
|| ',' || cgnv.INSTRUCTORS as updaters, cgnv.SECTION_ID, cgnv.TERM
from course_group_name_v cgnv
```

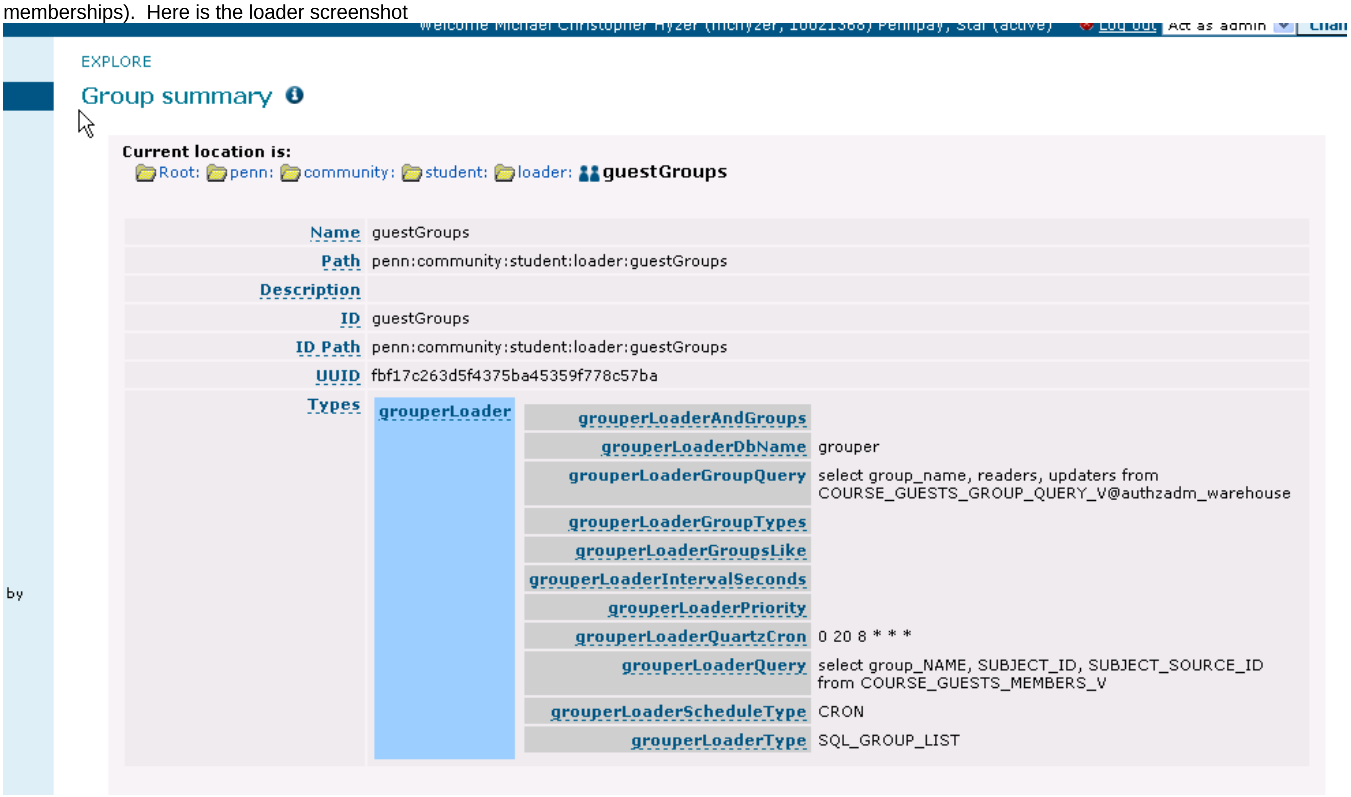
- Here is an example of data returned:

GROUP_NAME	READERS	UPDATERS	SECTION_ID	TERM
penn:community:student:course:2009C:AS:ANTH:258:401:assistants_systemOfRecord	penn:community:student:security:courseReaders, penn:community:student:course:2009C:AS:ANTH:258:401:assistants,penn:community:student:course:2009C:AS:ANTH:258:401:instructors	penn:community:student:security:courseUpdaters, penn:community:student:course:2009C:AS:ANTH:258:401:assistants,penn:community:student:course:2009C:AS:ANTH:258:401:instructors	ANTH258401	2009C
penn:community:student:course:2009C:FA:ARCH:727:401:assistants_systemOfRecord	penn:community:student:security:courseReaders, penn:community:student:course:2009C:FA:ARCH:727:401:assistants,penn:community:student:course:2009C:FA:ARCH:727:401:instructors	penn:community:student:security:courseUpdaters, penn:community:student:course:2009C:FA:ARCH:727:401:assistants,penn:community:student:course:2009C:FA:ARCH:727:401:instructors	ARCH727401	2009C
penn:community:student:course:2009C:EG:BE:099:001:assistants_systemOfRecord	penn:community:student:security:courseReaders, penn:community:student:course:2009C:EG:BE:099:001:assistants,penn:community:student:course:2009C:EG:BE:099:001:instructors	penn:community:student:security:courseUpdaters, penn:community:student:course:2009C:EG:BE:099:001:assistants,penn:community:student:course:2009C:EG:BE:099:001:instructors	BE 099001	2009C

-

Primary guests group

The guests group is an ad hoc group, but we want it auto-created, and managed by the loader for security. The grouper loader doesnt really support this at the moment (we should add support), but one workaround is just to sync the members from the grouper membership table (immediate memberships). Here is the loader screenshot



Current location is:
Root: penn: community: student: loader: guestGroups

Name	guestGroups
Path	penn:community:student:loader:guestGroups
Description	
ID	guestGroups
ID Path	penn:community:student:loader:guestGroups
UUID	fbf17c263d5f4375ba45359f778c57ba
Types	grouperLoader

grouperLoaderAndGroups	
grouperLoaderDbName	grouper
grouperLoaderGroupQuery	select group_name, readers, updaters from COURSE_GUESTS_GROUP_QUERY_V@authzadm_warehouse
grouperLoaderGroupTypes	
grouperLoaderGroupsLike	
grouperLoaderIntervalSeconds	
grouperLoaderPriority	
grouperLoaderQuartzCron	0 20 8 * * *
grouperLoaderQuery	select group_NAME, SUBJECT_ID, SUBJECT_SOURCE_ID from COURSE_GUESTS_MEMBERS_V
grouperLoaderScheduleType	CRON
grouperLoaderType	SQL_GROUP_LIST

- Note, this is driven from the grouper db connection, since it needs to hit the grouper registry for membership info. The members view (needs to join with a db link to the warehouse to get the group names), note: driving site means that the names should be brought from the warehouse to the grouper registry, then calculated there. It runs at 8:15 (even though screen says 8:20, this was changed). Note this query will need to change in 1.5

```
CREATE OR REPLACE VIEW COURSE_GUESTS_MEMBERS_V
(GROUP_NAME, SUBJECT_ID, SUBJECT_SOURCE_ID)
AS
select /*+DRIVING_SITE(gga)*/
cgnv.guests as group_name, gm.subject_id, gm.SUBJECT_SOURCE as subject_source_id
 from course_group_name_v@authzadm_warehouse cgnv,
grouper_attributes gga,
grouper_fields ggf, grouper_memberships gms, grouper_fields gf,
grouper_members gm
where gga.field_id = ggf.ID and ggf.NAME = 'name'
and gga.VALUE = cgnv.guests
and cgnv.primary_course = 'T'
and gms.OWNER_ID = gga.GROUP_ID
and gms.FIELD_ID = gf.ID
and gms.MSHIP_TYPE = 'immediate'
and gf.NAME = 'members'
and gms.MEMBER_ID = gm.id
```

- This gives data like this:

GROUP_NAME	SUBJECT_ID	SUBJECT_SOURCE_ID
penn:community:student:course:2009C:AS:ANTH:258:401:guests	12345678	pennperson
penn:community:student:course:2009C:AS:ANTH:258:402:guests	12345677	pennperson

- The groups query is run via db link as well (since the members query must be run from grouper registry, and both queries must run from same db). Here is the view (on warehouse)

```
CREATE OR REPLACE VIEW COURSE_GUESTS_GROUP_QUERY_V
(GROUP_NAME, READERS, UPDATERS)
AS
select primary_cgnv.guests as group_name,
'penn:community:student:security:courseReaders, ' || primary_cgnv.ASSISTANTS
|| ',' || primary_cgnv.INSTRUCTORS as readers,
'penn:community:student:security:courseUpdaters, ' || primary_cgnv.ASSISTANTS
|| ',' || primary_cgnv.INSTRUCTORS as updaters
from course_group_name_v primary_cgnv
where primary_cgnv.primary_course = 'T'
```

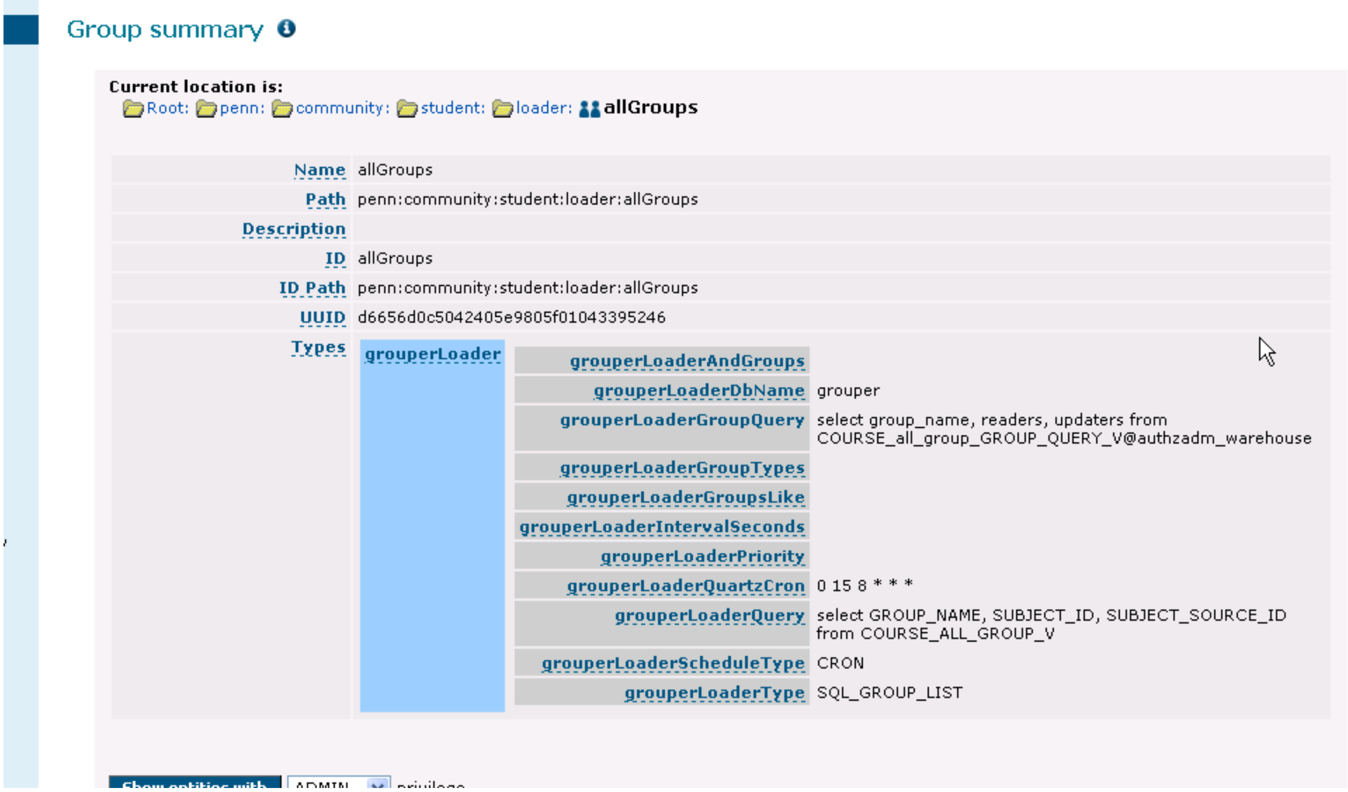
- Here is an example of the data:

GROUP_NAME	READERS	UPDATERS
penn:community:student:course:2009C:EG:BE:099:001:guests	penn:community:student:security:courseReaders, penn:community:student:course:2009C:EG:BE:099:001:assistants, penn:community:student:course:2009C:EG:BE:099:001:instructors	penn:community:student:security:courseUpdaters, penn:community:student:course:2009C:EG:BE:099:001:assistants, penn:community:student:course:2009C:EG:BE:099:001:instructors
penn:community:student:course:2009C:EG:BE:100:001:guests	penn:community:student:security:courseReaders, penn:community:student:course:2009C:EG:BE:100:001:assistants, penn:community:student:course:2009C:EG:BE:100:001:instructors	penn:community:student:security:courseUpdaters, penn:community:student:course:2009C:EG:BE:100:001:assistants, penn:community:student:course:2009C:EG:BE:100:001:instructors
penn:community:student:course:2009C:EG:BE:100:201:guests	penn:community:student:security:courseReaders, penn:community:student:course:2009C:EG:BE:100:201:assistants, penn:community:student:course:2009C:EG:BE:100:201:instructors	penn:community:student:security:courseUpdaters, penn:community:student:course:2009C:EG:BE:100:201:assistants, penn:community:student:course:2009C:EG:BE:100:201:instructors

-

The 'All' groups for primary courses

Now that we have the students, instructors, assistants and guests, we can roll up all of those into an 'all' group. Here is the loader screenshot:



Group summary

Current location is:
Root: penn: community: student: loader: **allGroups**

Name	allGroups
Path	penn:community:student:loader:allGroups
Description	
ID	allGroups
ID Path	penn:community:student:loader:allGroups
UUID	d6656d0c5042405e9805f01043395246
Types	grouperLoader

grouperLoader	
grouperLoaderAndGroups	
grouperLoaderDbName	grouper
grouperLoaderGroupQuery	select group_name, readers, updaters from COURSE_all_group_GROUP_QUERY_V@authzadm_warehouse
grouperLoaderGroupTypes	
grouperLoaderGroupsLike	
grouperLoaderIntervalSeconds	
grouperLoaderPriority	
grouperLoaderQuartzCron	0 15 8 * * *
grouperLoaderQuery	select GROUP_NAME, SUBJECT_ID, SUBJECT_SOURCE_ID from COURSE_ALL_GROUP_V
grouperLoaderScheduleType	CRON
grouperLoaderType	SQL_GROUP_LIST

- This runs at 8:20 (change from before, the screen says 8:15). Since this builds on existing groups and we need to know the group id for the subject id.
- Here is the membership view (in warehouse db):

```
CREATE OR REPLACE VIEW COURSE_ALL_GROUP_MEMBERS_V
(GROUP_NAME, MEMBER_GROUP_NAME)
AS
select primary_cgnv.allgroup as group_name,
primary_cgnv.ASSISTANTS as member_group_name
from course_group_name_v primary_cgnv
where primary_cgnv.primary_course = 'T'
union all
select primary_cgnv.allgroup as group_name,
primary_cgnv.guests as member_group_name
from course_group_name_v primary_cgnv
where primary_cgnv.primary_course = 'T'
union all
select primary_cgnv.allgroup as group_name,
primary_cgnv.students as member_group_name
from course_group_name_v primary_cgnv
where primary_cgnv.primary_course = 'T'
union all
select primary_cgnv.allgroup as group_name,
primary_cgnv.instructors as member_group_name
from course_group_name_v primary_cgnv
where primary_cgnv.primary_course = 'T'
```

- Here is the membership view (in grouper db): [note, this query will change in 1.5)

```
CREATE OR REPLACE VIEW COURSE_ALL_GROUP_V
(GROUP_NAME, SUBJECT_ID, SUBJECT_SOURCE_ID, MEMBER_GROUP_NAME)
AS
select /*+DRIVING_SITE(gga)*/
cagmv.GROUP_NAME,
gga.GROUP_ID as subject_id, 'g:gsa' as subject_source_id,
cagmv.member_group_name
from COURSE_ALL_GROUP_MEMBERS_V@authzadm_warehouse cagmv, grouper_attributes gga,
grouper_fields ggf
where gga.field_id = ggf.ID and ggf.NAME = 'name'
and gga.VALUE = cagmv.member_group_name
```

- Here is an example of the data returned:

GROUP_NAME	SUBJECT_ID	SUBJECT_SOURCE_ID	MEMBER_GROUP_NAME
penn:community:student:course:2009C:EG:BE:099:001:all	f73ec967fd7f4710929a82f5d9d299cf	g:gsa	penn:community:student:course:2009C:EG:BE:099:001:assistants
penn:community:student:course:2009C:EG:BE:099:001:all	28374298347928347982374928347	g:gsa	penn:community:student:course:2009C:EG:BE:099:001:guests
penn:community:student:course:2009C:EG:BE:099:001:all	287349823749827349827349823749	g:gsa	penn:community:student:course:2009C:EG:BE:099:001:students
penn:community:student:course:2009C:EG:BE:099:001:all	2387234897234987293874982374	g:gsa	penn:community:student:course:2009C:EG:BE:099:001:instructors
penn:community:student:course:2009C:EG:BE:099:002:all	23874928374982374982734897329	g:gsa	penn:community:student:course:2009C:EG:BE:099:002:assistants

- Here is the group view

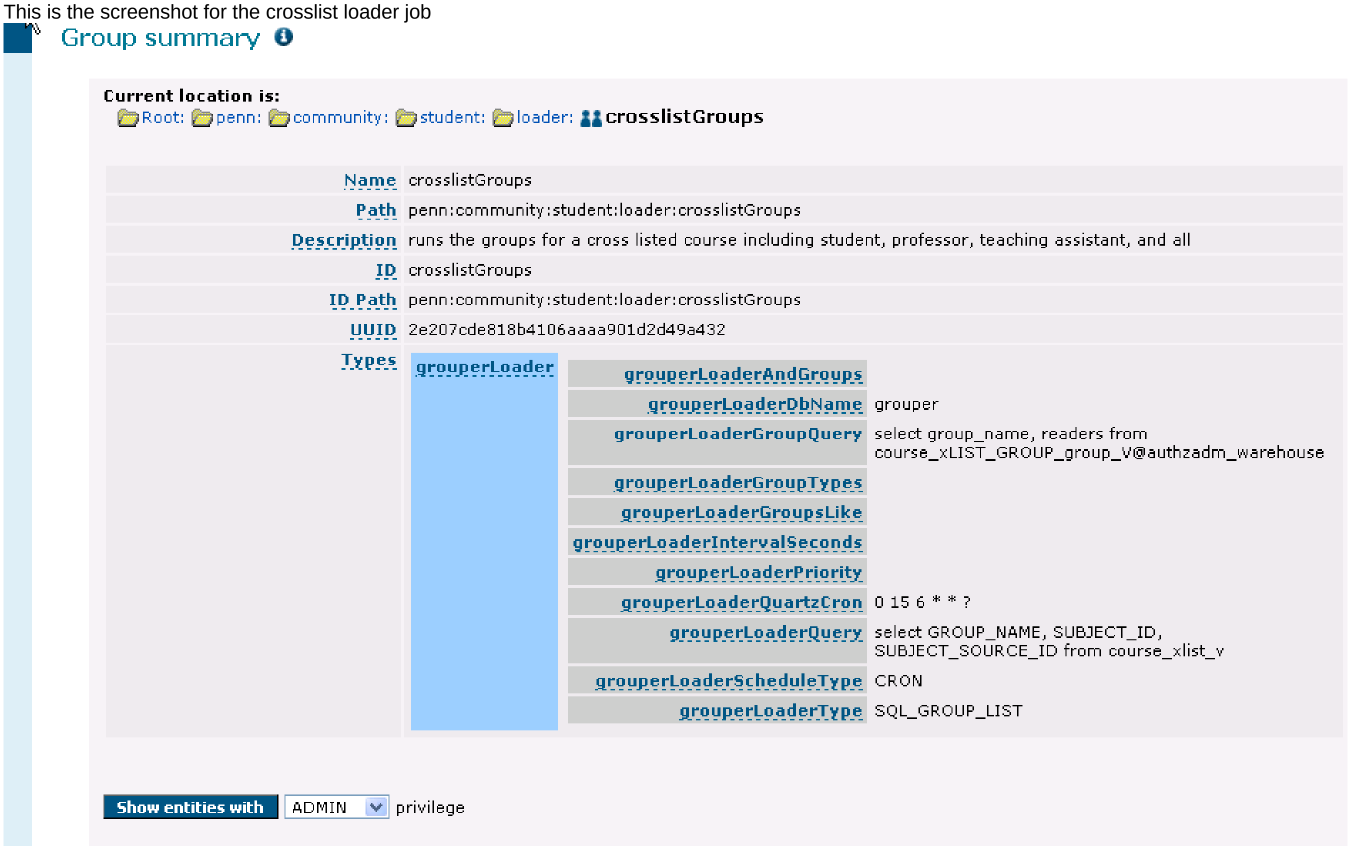
```
CREATE OR REPLACE VIEW COURSE_ALL_GROUP_GROUP_QUERY_V
(GROUP_NAME, READERS, UPDATERS)
AS
select primary_cgnv.ALLGROUP as group_name,
'penn:community:student:security:courseReaders, ' || primary_cgnv.ASSISTANTS
|| ',' || primary_cgnv.INSTRUCTORS as readers,
null as updaters
from course_group_name_v primary_cgnv
where primary_cgnv.primary_course = 'T'
```

- Here is an example of the data

GROUP_NAME	READERS	UPDATERS
penn:community:student:course:2009C:EG:BE:099:001:all	penn:community:student:security:courseReaders, penn:community:student:course:2009C:EG:BE:099:001:assistants,penn:community:student:course:2009C:EG:BE:099:001:instructors	
penn:community:student:course:2009C:EG:BE:100:001:all	penn:community:student:security:courseReaders, penn:community:student:course:2009C:EG:BE:100:001:assistants,penn:community:student:course:2009C:EG:BE:100:001:instructors	

Crosslisted course lists

This is the screenshot for the crosslist loader job



Group summary

Current location is:
Root: penn: community: student: loader: crosslistGroups

Name	crosslistGroups
Path	penn:community:student:loader:crosslistGroups
Description	runs the groups for a cross listed course including student, professor, teaching assistant, and all
ID	crosslistGroups
ID Path	penn:community:student:loader:crosslistGroups
UUID	2e207cde818b4106aaaa901d2d49a432
Types	grouperLoader

grouperLoader	
grouperLoaderAndGroups	
grouperLoaderDbName	grouper
grouperLoaderGroupQuery	select group_name, readers from course_xLIST_GROUP_group_V@authzadm_warehouse
grouperLoaderGroupTypes	
grouperLoaderGroupsLike	
grouperLoaderIntervalSeconds	
grouperLoaderPriority	
grouperLoaderQuartzCron	0 15 6 * * ?
grouperLoaderQuery	select GROUP_NAME, SUBJECT_ID, SUBJECT_SOURCE_ID from course_xlist_v
grouperLoaderScheduleType	CRON
grouperLoaderType	SQL_GROUP_LIST

- Again, since this is loading groups in other groups, we need to run from the grouper DB connection. It says 6:15, but this is really run at 8:30am.
- Here is the cross list members view in the warehouse:

```
CREATE OR REPLACE VIEW COURSE_XLIST_GROUP_MEMBERS_V
(GROUP_NAME, MEMBER_GROUP_NAME)
AS
select xlist_cgnv.ASSISTANTS as group_name,
primary_cgnv.ASSISTANTS as member_group_name
from course_group_name_v xlist_cgnv,
course_group_name_v primary_cgnv
where xlist_cgnv.PRIMARY_COURSE = 'F'
and primary_cgnv.primary_course = 'T'
and xlist_cgnv.XLIST_PRIMARY = primary_cgnv.SECTION_ID
union all
select xlist_cgnv.instructors as group_name,
primary_cgnv.instructors as member_group_name
from course_group_name_v xlist_cgnv,
course_group_name_v primary_cgnv
where xlist_cgnv.PRIMARY_COURSE = 'F'
and primary_cgnv.primary_course = 'T'
and xlist_cgnv.XLIST_PRIMARY = primary_cgnv.SECTION_ID
union all
select xlist_cgnv.students as group_name,
primary_cgnv.students as member_group_name
from course_group_name_v xlist_cgnv,
course_group_name_v primary_cgnv
where xlist_cgnv.PRIMARY_COURSE = 'F'
and primary_cgnv.primary_course = 'T'
and xlist_cgnv.XLIST_PRIMARY = primary_cgnv.SECTION_ID
union all
select xlist_cgnv.ALLGROUP as group_name,
primary_cgnv.allgroup as member_group_name
from course_group_name_v xlist_cgnv,
course_group_name_v primary_cgnv
where xlist_cgnv.PRIMARY_COURSE = 'F'
```

```
and primary_cgnv.primary_course = 'T'
and xlist_cgnv.XLIST_PRIMARY = primary_cgnv.SECTION_ID
union all
select xlist_cgnv.GUESTS as group_name,
primary_cgnv.guests as member_group_name
from course_group_name_v xlist_cgnv,
course_group_name_v primary_cgnv
where xlist_cgnv.PRIMARY_COURSE = 'F'
and primary_cgnv.primary_course = 'T'
and xlist_cgnv.XLIST_PRIMARY = primary_cgnv.SECTION_ID
```

- Here is the members query in the grouper database

```
CREATE OR REPLACE VIEW COURSE_XLIST_V
(GROUP_NAME, SUBJECT_ID, SUBJECT_SOURCE_ID, XLIST_PRIMARY_GROUP_NAME)
AS
select cgm.GROUP_NAME,
ga.GROUP_ID as subject_id, 'g:gsa' as subject_source_id,
cgm.member_group_name as xlist_primary_group_name
from course_xLIST_GROUP_MEMBERS_V@authzadm_warehouse cgm, grouper_attributes ga,
grouper_fields gf
where ga.field_id = gf.ID and gf.NAME = 'name'
and ga.VALUE = cgm.member_group_name
```

- Here is a sample of the data returned

GROUP_NAME	SUBJECT_ID	SUBJECT_SOURCE_ID	XLIST_PRIMARY_GROUP_NAME
penn:community:student:course:2009C:AS:ANTH:258:401:assistants	dc0b03453453453454357d7d23c4	g:gsa	penn:community:student:course:2009C:EG:CIS:106:401:assistants
penn:community:student:course:2009C:FA:ARCH:727:401:all	234234234234234234234234234	g:gsa	penn:community:student:course:2009C:EG:IPD:527:401:all
penn:community:student:course:2009C:EG:BE:330:401:instructors	435345345345345345345435345	g:gsa	penn:community:student:course:2009C:EG:MSE:330:401:instructors

- Here is the all group query

```
CREATE OR REPLACE VIEW COURSE_XLIST_GROUP_GROUP_V
(GROUP_NAME, READERS)
AS
select xlist_cgnv.ASSISTANTS as group_name,
'penn:community:student:security:courseReaders, ' || primary_cgnv.ASSISTANTS
|| ',' || primary_cgnv.INSTRUCTORS as readers
from course_group_name_v xlist_cgnv,
course_group_name_v primary_cgnv
where xlist_cgnv.PRIMARY_COURSE = 'F'
and primary_cgnv.primary_course = 'T'
and xlist_cgnv.XLIST_PRIMARY = primary_cgnv.SECTION_ID
union all
select xlist_cgnv.INSTRUCTORS as group_name,
'penn:community:student:security:courseReaders, ' || primary_cgnv.ASSISTANTS
|| ',' || primary_cgnv.INSTRUCTORS as readers
from course_group_name_v xlist_cgnv,
course_group_name_v primary_cgnv
where xlist_cgnv.PRIMARY_COURSE = 'F'
and primary_cgnv.primary_course = 'T'
and xlist_cgnv.XLIST_PRIMARY = primary_cgnv.SECTION_ID
union all
select xlist_cgnv.STUDENTS as group_name,
'penn:community:student:security:courseReaders, ' || primary_cgnv.ASSISTANTS
|| ',' || primary_cgnv.INSTRUCTORS as readers
from course_group_name_v xlist_cgnv,
course_group_name_v primary_cgnv
where xlist_cgnv.PRIMARY_COURSE = 'F'
and primary_cgnv.primary_course = 'T'
and xlist_cgnv.XLIST_PRIMARY = primary_cgnv.SECTION_ID
union all
select xlist_cgnv.allgroup as group_name,
'penn:community:student:security:courseReaders, ' || primary_cgnv.ASSISTANTS
|| ',' || primary_cgnv.INSTRUCTORS as readers
from course_group_name_v xlist_cgnv,
course_group_name_v primary_cgnv
where xlist_cgnv.PRIMARY_COURSE = 'F'
and primary_cgnv.primary_course = 'T'
and xlist_cgnv.XLIST_PRIMARY = primary_cgnv.SECTION_ID
union all
select xlist_cgnv.guests as group_name,
'penn:community:student:security:courseReaders, ' || primary_cgnv.ASSISTANTS
|| ',' || primary_cgnv.INSTRUCTORS as readers
from course_group_name_v xlist_cgnv,
course_group_name_v primary_cgnv
where xlist_cgnv.PRIMARY_COURSE = 'F'
and primary_cgnv.primary_course = 'T'
and xlist_cgnv.XLIST_PRIMARY = primary_cgnv.SECTION_ID
```

- Here is sample data:

GROUP_NAME	READERS
penn:community:student:course:2009C:AS:ANTH:258:401:assistants	penn:community:student:security:courseReaders, penn:community:student:course:2009C:EG:CIS:106:401:assistants,penn:community:student:course:2009C:EG:CIS:106:401:instructors
penn:community:student:course:2009C:FA:ARCH:727:401:instructors	penn:community:student:security:courseReaders, penn:community:student:course:2009C:EG:IPD:527:401:assistants,penn:community:student:course:2009C:EG:IPD:527:401:instructors

-