

Grouper membership eligibility requirements

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

This applies to Grouper v2.6.16+

To enforce membership eligibility, you can use a composite, rules, JEXL scripted groups, or you can use this new feature. You can link an attribute with an eligibility group so that immediate memberships (not effective, composite, loaded) from a (probably) manual group will be veto-ed or removed when users are no longer eligible because they are no longer effectively in the eligible population group.

Steps to implement


These are explained below

1. Identify coarse grained populations that manual groups could be constrained to
2. Create attribute def(s) and attribute names.
 - a. Make as many attribute defs as you want (equal to or less than the number of populations)
 - b. The number of attribute names is 1-to-1 to the coarse grained populations.
3. Allow certain groups to be able to read and assign the attribute (e.g. power users)
4. Configure the attribute in grouper.properties to be linked to an eligibility group (e.g. employees)
5. Configure the veto text in the externalized text file
6. Configure the attribute to be assigned in the group edit screen (optional)
 - a. We can allow attributes on stem edit screens in future
7. Assign the attribute to groups or folders
 - a. Note, loader groups are not affected
8. Membership hook will veto membership adds if the user is not eligible
9. Change log consumer will remove members when no longer eligible
10. Full sync daemon will make sure everything is correct
11. When members are removed a record is kept in the grouper_mship_req_change table
 - a. A simple GSH script could easily rollback changes made by this module

Identify coarse grained populations

Note, these do not need to be direct members.

[Home](#) > [Root](#) > [ref](#) > [employee](#)



employee

[+ Add members](#)

[Group actions ▾](#)

[Show details ▾](#)

Members

Privileges

More ▾

The following table lists all entities which are members of this group.

Filter for:

All members ▾




Member name

Apply filter

Reset

Advanced

Remove selected members

<input type="checkbox"/> Entity name ▾	Membership	Choose action
<input type="checkbox"/>  my name is test.subject.2	Direct	Actions ▾
<input type="checkbox"/>  my name is test.subject.3	Direct	Actions ▾
<input type="checkbox"/>  my name is test.subject.4	Direct	Actions ▾

Show: 100 ▾

Showing 1-3 of 3 · [First](#) | [Prev](#) | [Next](#) | [Last](#)

Create attributes

These settings need to be exactly like this. Needs to not be multi-assignable or have a value...

membershipRequirementDef

Edit attribute definition

Attribute definition ID:

membershipRequirementDef

ID is the unique identifier for this attribute definition. It should be short and simple, and might have character restrictions. The ID should rarely change, if ever.

Description:

Description contains notes about the attribute definition, which could include: what the attribute definition represents, why it was created, etc.

Type:

Attribute

Attribute definition type describes the attribute definition. Generally it will be attribute or permission. Type is used for templates, limit describes a permission, and service identifies which application the object refers to.

Assign to:

- | | |
|---|---|
| <input type="checkbox"/> Attribute definition | <input type="checkbox"/> Attribute definition attribute assignment |
| <input checked="" type="checkbox"/> Folder | <input type="checkbox"/> Folder attribute assignment |
| <input checked="" type="checkbox"/> Group / Role / Local entity | <input type="checkbox"/> Group / Role / Local entity attribute assignment |
| <input type="checkbox"/> Member | <input type="checkbox"/> Member attribute assignment |
| <input type="checkbox"/> Membership | <input type="checkbox"/> Membership attribute assignment |
| <input type="checkbox"/> Membership - immediate only | <input type="checkbox"/> Membership - immediate only - attribute assignment |

Designate which types of objects that this definition can be assigned to. There are six base object types, or you can assign attributes to the assignment of attributes to those base object types. Membership can be assigned to an immediate or an effective membership, and will still exist as an orphan if the membership is unassigned until the membership is reassigned. Immediate membership attribute assignments are only assignable to immediate memberships and are automatically deleted once the membership is unassigned.

Multi-assignable:

☐

If this attribute can be assigned to the same owner object more than once. For instance, a Group can have more than one Rule attached to it, so the Rule attribute is multi-assignable

Value type:

No value

If this attribute assignment holds one or more values, this is the type. If there are no allowed values, select No value.

Multi-valued:

☐

If this attribute has values, if it can have more than one value assigned at once.

[Show advanced properties](#) ▾

Home > New attribute name

New attribute name

Attribute definition: etc:attribute:membershipRequirement:membershipRequirementDef

The attribute definition holds the settings and security for attribute. Each attribute definition can have multiple attribute names. Every attribute name is associated with one and only one attribute definition.

Folder: etc:attribute:membershipRequirement

Enter a folder name or [search for a folder where you are allowed to create new attribute def names.](#)

Name of attribute name: requireEmployee *

Name is the label that identifies this attribute name, and might change.

ID of attribute name: requireEmployee

☐ Edit the ID *

ID is the unique identifier you set for this attribute name. The ID must be unique within this folder, and should rarely change. It can be used by other systems to refer to this attribute name. The ID field cannot contain spaces or special characters.

Description:

Description contains notes about the attribute name, which could include: what the attribute name represents, why it was created, etc.

Save

Cancel

Allow certain people to be able to read and assign the attribute (e.g. power users)

Home > Root > etc > attribute > membershipRequirement > membershipRequirementDef

Attribute definition

+ Add members
Attribute actions

membershipRequirementDef

Member name or ID:
Enter an entity name or ID, or [search for an entity](#).

Assign these privileges:

☐ ADMIN
☒ UPDATE
☒ READ
☐ VIEW
☐ OPTIN
☐ OPTOUT
☐ ATTRIBUTE READ
☐ ATTRIBUTE UPDATE

Add

Show details

Attribute names
Privileges
More

The following table lists all entities with privileges for this attribute definition.

Filter for:

Apply filter
Reset
Advanced

Update:
Assign the ADMIN privilege
Update selected

<input type="checkbox"/> Entity name	Admin	Read	Update	OptIn	OptOut	Attribute read	Attribute update	View	Choose action
<input type="checkbox"/> my name is test.subject.3		✓	✓	✓	✓			✓	Actions
<input type="checkbox"/> my name is test.subject.4		✓	✓	✓	✓			✓	Actions
<input type="checkbox"/> powerUsers		✓	✓	✓	✓			✓	Actions

Show:
100

Showing 1-3 of 3 · First | Prev | Next | Last

Configure the attribute in grouper.properties to be linked to an eligibility group (e.g. employees)

```
# ui key to externalize text (error message)
grouper.membershipRequirement.requireEmployee.uiKey = vetoRequireEmployee

# attribute name that signifies this requirement
grouper.membershipRequirement.requireEmployee.attributeName = etc:attribute:membershipRequirement:requireEmployee

# group name which is the population group
grouper.membershipRequirement.requireEmployee.requireGroupName = ref:employee

# if the overall hook is enabled, is the hook for this specific config enabled? defaults to true.
grouper.membershipRequirement.requireEmployee.hookEnable = true
```

Here is the base config which has general settings

```
#####
## Custom veto composites membership requirement
## This feature allows users to auto-veto ineligible members or remove them when they become ineligible.
## Note that each custom composite also needs to be defined in the Grouper UI text properties in order
## to provide a friendly description in the UI.  customCompositeMinusEmployees and customCompositeIntersectIt
are also defined as examples there.
#####

# how long should logs of membership requirement logs be stored in database?
grouper.membershipRequirement.keepLogsForDays = 90

# should hook for membership veto be enabled
grouper.membershipRequirement.hookEnable = true

# should changeLog for membership veto change log be enabled in general
grouper.membershipRequirement.changeLogEnable = true

# ui key to externalize text
#grouper.membershipRequirement.someConfigId.uiKey = customVetoCompositeRequireEmployee

# attribute name that signifies this requirement
#grouper.membershipRequirement.someConfigId.attributeName = etc:attribute:customComposite:requireEmployee

# group name which is the population group
#grouper.membershipRequirement.someConfigId.requireGroupName = org:centralIt:staff:itStaff

# if the overall hook is enabled, is the hook for this specific config enabled? defaults to true.
#grouper.membershipRequirement.someConfigId.hookEnable = true
```

Configure the veto text in the externalized text file

"grouper.text.en.us.properties" (veto.membershipVeto.customComposite.%uiKey% = %text you want displayed as error%)

```
veto.membershipVeto.customComposite.vetoRequireEmployee = Only employees can be members of this group
```

Configure the attribute to be assigned in the group edit screen (optional)

grouper.properties

```
groupScreen.attribute.requireEmployee.attributeName = etc:attribute:membershipRequirement:requireEmployee
groupScreen.attribute.requireEmployee.label = Require employee
groupScreen.attribute.requireEmployee.description = Members of this group (or groups in folder) will be
required to be employees, otherwise they will be vetoed or removed
groupScreen.attribute.requireEmployee.index = 1
```

Assign the attribute to groups or folders

Note, this will remove direct members. Note, be careful about service principals, maybe those need to be in a different manual group?

Groups:

+ Create new group

Quick links

My groups

My folders

My favorites

My services

My activity

Miscellaneous

Browse folders

Root

app

confluence

policy

financialSystem

studentSystem

etc

attribute

abacJexlScript

attestation

attrExternalSubjectInvite

attrLoader

attributeAutoCreate

customUi

entities

instrumentationData

loaderLdap

loaderMetadata

membershipRequirement

membershipRequirementC

requireEmployee

Home > Root > app > confluence > policy > financialSystem

financialSystem

Edit group

Group name: financialSystem

Name is the label that identifies this group, and might change.

Group ID: financialSystem

ID is the unique identifier for this group. It should be short and simple, and might have character restrictions. The ID should rarely change, if ever.

Alternate ID path:

Alternate ID path allows groups to be searchable using an alternate name. The format is the same as the format of ID path.

Description:

Description contains notes about the group, which could include: what the group represents, why it was created, etc.

Enabled date: yyyy/mm/dd hh:mi am/pm

When this group will be enabled if the time is in the future. Expected timezone is ET.

Disabled date: yyyy/mm/dd hh:mi am/pm

When this group will be disabled if the time is in the future. Expected timezone is ET.

Require employee: ☐

Members of this group (or groups in folder) will be required to be employees, otherwise they will be vetoed or removed

Show advanced properties

Save Cancel

Folders

+ Create new group

Quick links

My groups

My folders

My favorites

My services

My activity

Miscellaneous

Browse folders

Root

app

confluence

policy

financialSystem

studentSystem

etc

attribute

abacJexlScript

attestation

attrExternalSubjectInvite

attrLoader

attributeAutoCreate

Home > Root > app > confluence > policy

policy

Show details

Edit folder

Folder actions

Folder contents

Privileges

Attribute Assignments

+ Assign attribute

The following table lists all attributes assigned to this folder

Attribute name: etc:attribute:membershipRequirement:requireEmployee

The attribute name is the part of the attribute which is assigned to owner objects. Generally multiple attribute names are related to one attribute definition.

Save

Assignment type	Attribute name	Enabled?	Assignment values	Attribute definition	Choose action
Direct assignment	requireEmployee	enabled		membershipRequirementDef	Actions

Membership hook will veto membership adds if the user is not eligible

Only employees can be members of this group

Create new group

Quick links

My groups

My folders

My favorites

My services

My activity

Miscellaneous

Browse folders

Root

app

confluence

policy

financialSystem

studentSystem

etc

attribute

abacJexlScript

attestation

attrExternalSubjectInvite

attrLoader

attributeAutoCreate

customUi

entities

instrumentationData

loaderLdap

loaderMetadata

membershipRequirement

membersioRequirementC

Home > Root > app > confluence > policy > financialSystem

financialSystem

Add members

Group actions

Member name or ID:

description.test.subject.8

Enter an entity name or ID, or [search for an entity.](#)

Assign these privileges:

Default privileges

Custom privileges

Start date:

yyyy/mm/dd hh:mi am/pm

The optional date on which this entity's membership begins. Expected timezone is ET.

End date:

yyyy/mm/dd hh:mi am/pm

The optional date on which this entity's membership expires. Expected timezone is ET.

Add

 or import a list of members .

Show details

Members

Privileges

More

The following table lists all entities which are members of this group.

Filter for:

All members

Member name

Apply filter

Reset

Advanced

Remove selected members

<input type="checkbox"/>	Entity name	Membership	Choose action
<input type="checkbox"/>	my name is test.subject.0	Direct	Actions
<input type="checkbox"/>	my name is test.subject.2	Direct	Actions

Show:

100

Showing 1-2 of 2 · First | Prev | Next | Last

Change log consumer will remove members when no longer eligible

Home > Miscellaneous > All daemon jobs > Daemon logs

Daemon logs

Job actions ▾

Filter for:CHANGE_LOG_consumer_membershipRequire

Start time between:

yyyy-mm-dd hh:mi:ss

yyyy-mm-dd hh:mi:ss

End time between:

yyyy-mm-dd hh:mi:ss

yyyy-mm-dd hh:mi:ss

Last updated between:

yyyy-mm-dd hh:mi:ss

yyyy-mm-dd hh:mi:ss

Subjobs:

☐ Show subjobs

Status:

☐ Success ☐ Error ☐ Started ☐ Running ☐ Warning ☐ Config error ☐ Subject problems

Number of rows:

400

Apply filter

Reset

64 logs found for job name: CHANGE_LOG_consumer_membershipRequire

Status	Loaded group	Job type	Start time	End time	Millis	Millis get data	Millis load data	Total count	Add count	Update count	Delete count	Unresolvable count	Log ID
Success	N/A	overall	2022-08-29 13:02:00.0	2022-08-29 13:02:00.0	244			1	0	0	2	0	2c97808182e
Success	N/A	overall	2022-08-29 13:01:00.0	2022-08-29 13:01:00.0	19			0	0	0	0	0	2c97808182e
Success	N/A	overall	2022-08-29 13:00:00.0	2022-08-29 13:00:00.0	15			0	0	0	0	0	2c97808182e

Full sync daemon will make sure everything is correct

Home > Miscellaneous > All daemon jobs > Daemon logs

Daemon logs

Job actions ▾

Filter for:OTHER_JOB_grouperMembershipRequireFull

Start time between:

yyyy-mm-dd hh:mi:ss

yyyy-mm-dd hh:mi:ss

End time between:

yyyy-mm-dd hh:mi:ss

yyyy-mm-dd hh:mi:ss

Last updated between:

yyyy-mm-dd hh:mi:ss

yyyy-mm-dd hh:mi:ss

Subjobs:

☐ Show subjobs

Status:

☐ Success ☐ Error ☐ Started ☐ Running ☐ Warning ☐ Config error ☐ Subject problems

Number of rows:

400

Apply filter

Reset

4 logs found for job name: OTHER_JOB_grouperMembershipRequireFull

Status	Loaded group	Job type	Start time	End time	Millis	Millis get data	Millis load data	Total count	Add count	Update count	Delete count	Unresolvable count	Log ID
Success	N/A	overall	2022-08-29 15:43:20.0	2022-08-29 15:43:21.0	941			0	0	0	2	0	2c9780818;
Success	N/A	overall	2022-08-29 15:42:55.0	2022-08-29 15:42:56.0	911			0	0	0	2	0	2c9780818;
Success	N/A	overall	2022-08-29 13:52:31.0	2022-08-29 13:52:32.0	745			0	0	0	0	0	2c9780818;

When members are removed a record is kept in the grouper_mship_req_change table

the_timestamp	removed_from	description	eligibility_group	attribute_name	config_id	engine
2022-08-29 16:06:05.0	app:confluence:policy:financialSystem	description.test.subject.2	ref:employee	etc:attribute:membershipRequirement:requireEmployee	requireEmployee	changeLog
2022-08-29 16:06:01.0	app:confluence:policy:studentSystem	description.test.subject.2	ref:employee	etc:attribute:membershipRequirement:requireEmployee	requireEmployee	changeLog
2022-08-29 15:43:21.0	app:confluence:policy:studentSystem	description.test.subject.2	ref:employee	etc:attribute:membershipRequirement:requireEmployee	requireEmployee	fullDaemon
2022-08-29 15:43:21.0	app:confluence:policy:financialSystem	description.test.subject.2	ref:employee	etc:attribute:membershipRequirement:requireEmployee	requireEmployee	fullDaemon
2022-08-29 15:42:56.0	app:confluence:policy:studentSystem	description.test.subject.0	ref:employee	etc:attribute:membershipRequirement:requireEmployee	requireEmployee	fullDaemon
2022-08-29 15:42:56.0	app:confluence:policy:financialSystem	description.test.subject.0	ref:employee	etc:attribute:membershipRequirement:requireEmployee	requireEmployee	fullDaemon
2022-08-29 13:02:00.0	app:confluence:policy:studentSystem	description.test.subject.8	ref:employee	etc:attribute:membershipRequirement:requireEmployee	requireEmployee	changeLog
2022-08-29 13:02:00.0	app:confluence:policy:financialSystem	description.test.subject.0	ref:employee	etc:attribute:membershipRequirement:requireEmployee	requireEmployee	changeLog

```

select
  gmrc.the_timestamp,
  gg.name as removed_from,
  gm.description,
  gg_elig.name as eligibility_group,
  gadn.name as attribute_name,
  gmrc.config_id,
  case
    when gmrc.engine = 'F' then 'fullDaemon'
    when gmrc.engine = 'C' then 'changeLog'
    else gmrc.engine
  end as engine
from
  grouper_mship_req_change gmrc,
  grouper_members gm,
  grouper_groups gg,
  grouper_groups gg_elig,
  grouper_attribute_def_name gadn
where
  gmrc.member_id = gm.id
  and gmrc.group_id = gg.id
  and gmrc.attribute_def_name_id = gadn.id
  and gmrc.require_group_id = gg_elig.id
order by
  the_timestamp desc;

```

TO DO

More features can be added to this:

1. Notifications (to managers or users)
2. Grace periods
3. Read-only mode
4. Exclude groups which are "exclude" type (doesn't exist yet)
5. Exclude groups by regex
6. Include only manual groups
7. Constrain subject sources
8. Remove when membership remove in folder (e.g. job or title changes)
9. Loader can restrict ineligible members
 - a. Confirm that JEXL scripted groups are affected correctly