

# Grouper rules pattern - Remove invalid membership due to group

<a href="#">Wiki Home</a>	<a href="#">Grouper Release Announcements</a>	<a href="#">Grouper Guides</a>	<a href="#">Grouper Deployment Guide</a>	<a href="#">Community Contributions</a>	<a href="#">Internal Developer Resources</a>
---------------------------	---	--------------------------------	--	---	--

## Grouper rules

If an entity is no longer a member of the employee group, remove them from the group for application X.

Note, you can key off of membershipRemove or flattenedMembershipRemove

Assign this rule to the group where the membership should be removed.

## Configure rule for v5+

The screenshot shows the Grouper web interface for configuring a rule. The breadcrumb trail is 'Home > Root > test > test1'. The page title is 'test1' with a 'Group actions' dropdown. Below the title is a 'Show details' link and tabs for 'Members', 'Privileges', and 'More'. The 'Group rules' section has a 'Rules actions' dropdown. A 'Documentation' link is present. The rule configuration form includes: 'Rule pattern' set to 'Remove invalid membership due to group' with a red asterisk; 'What action the rule should check for' set to 'Check if removed from group' with a red asterisk and a note: 'If you are checking a required group like employee then check if removed from group. If you are checking a forbidden group like lock-out then check if added to group.'; 'Group name to check' set to 'test2:test3:test4:group4' with a red asterisk and a note: 'Enter the group name of the group to check. Note this is the UI ID path, e.g. a:b:c'; and 'Run daemon override' set to a dropdown. At the bottom are 'Save' and 'Cancel' buttons.

Home > Root > test > test1

test1 Group actions ▾

Show details ▾

Members Privileges More ▾

Group rules Rules actions ▾

Documentation

Rule pattern Remove invalid membership due to group \*  
Type of rule that solves an existing use case with a certain action, condition, and result.

What action the rule should check for Check if removed from group \*  
If you are checking a required group like employee then check if removed from group. If you are checking a forbidden group like lock-out then check if added to group.

Group name to check test2:test3:test4:group4 \*  
Enter the group name of the group to check. Note this is the UI ID path, e.g. a:b:c

Run daemon override ▾  
If the rule has the ability to be run by daemon (for example to clean up data corruption), then you can set this to false if you do not want the daemon to run for this rule. If the rule does not have the ability to run in the daemon then this setting is ignored.

Save Cancel

## Configure rule for v4 and previous

### Java example

```

AttributeAssign attributeAssign = ruleGroup
    .getAttributeDelegate().addAttribute(RuleUtils.ruleAttributeDefName()).getAttributeAssign();

AttributeValueDelegate attributeValueDelegate = attributeAssign.getAttributeValueDelegate();

attributeValueDelegate.assignValue(
    RuleUtils.ruleActAsSubjectSourceIdName(), actAs.getSourceId());
attributeValueDelegate.assignValue(
    RuleUtils.ruleActAsSubjectIdName(), actAs.getId());

//note "mustBeInGroup" is the group (e.g. employees)
attributeValueDelegate.assignValue(
    RuleUtils.ruleCheckOwnerIdName(), mustBeInGroup.getId());
attributeValueDelegate.assignValue(
    RuleUtils.ruleCheckTypeName(),
    RuleCheckType.flattenedMembershipRemove.name());
attributeValueDelegate.assignValue(
    RuleUtils.ruleIfConditionEnumName(),
    RuleIfConditionEnum.thisGroupHasImmediateEnabledMembership.name());
attributeValueDelegate.assignValue(
    RuleUtils.ruleThenEnumName(),
    RuleThenEnum.removeMemberFromOwnerGroup.name());

//should be valid
String isValidString = attributeValueDelegate.retrieveValueString(
    RuleUtils.ruleValidName());

if (!StringUtils.equals("T", isValidString)) {
    throw new RuntimeException(isValidString);
}

```

## GSH shorthand method

```
RuleApi.groupIntersection(subjectActAs, ruleGroup, mustBeInGroup)
```

## GSH test case

```

gsh 0% grouperSession = GrouperSession.startRootSession();
edu.internet2.middleware.grouper.GrouperSession: d711e17ed44842a68b885bca5f294ab3,'GrouperSystem','application'
gsh 1% groupA = new GroupSave(grouperSession).assignName("stem:a").assignCreateParentStemsIfNotExist(true).
save();
group: name='stem:a' displayName='stem:a' uuid='4bc47ab6a6704132a73a31d34b83164b'
gsh 2% groupB = new GroupSave(grouperSession).assignName("stem:b").assignCreateParentStemsIfNotExist(true).
save();
group: name='stem:b' displayName='stem:b' uuid='22c410c494934a3baff8555940853ad1'
gsh 3% subjectActAs = SubjectFinder.findByIdAndSource("GrouperSystem", "g:isa", true);
subject: id='GrouperSystem' type='application' source='g:isa' name='GrouperSysAdmin'
gsh 4% RuleApi.groupIntersection(subjectActAs, groupA, groupB);
gsh 5% addMember("stem:a", "test.subject.0");
true
gsh 6% addMember("stem:b", "test.subject.0");
true
gsh 7% delMember("stem:b", "test.subject.0");
true
gsh 8% hasMember("stem:a", "test.subject.0");
false
gsh 9%

```

## GSH daemon test case

Run the above test case, then continue below:

```
gsh 9% addMember("stem:a", "test.subject.0");
true
gsh 10% status = GrouperLoader.runOnceByJobName(grouperSession, GrouperLoaderType.GROUPER_RULES);
loader ran successfully: Ran rules daemon, changed 0 records
gsh 11% hasMember("stem:a", "test.subject.0");
false
```

## Real world example

There is an includes list for IT staff at Penn. But anyone in that list must be an active employee or health system employee.

```
gsh 0% grouperSession = GrouperSession.startRootSession();
edu.internet2.middleware.grouper.GrouperSession: 6d8c3f9ea4c64e569e8d1bb292e989d1,'GrouperSystem','application'
gsh 1% itStaff_includes = GroupFinder.findByName(grouperSession, "penn:community:employee:itStaff_includes");
group: name='penn:community:employee:itStaff_includes' displayName='penn:community:employee:itStaff_includes'
uuid='59cd11b40d49446099e0409f755d9679'
gsh 2% employeeIncludingUphs = GroupFinder.findByName(grouperSession, "penn:community:employeeIncludingUphs");
group: name='penn:community:employeeIncludingUphs' displayName='penn:community:employeeIncludingUphs'
uuid='b0758e19dcd4431798cd5bfcfeb6ea66'
gsh 3% subjectActAs = SubjectFinder.findByIdAndSource("GrouperSystem", "g:isa", true);
subject: id='GrouperSystem' type='application' source='g:isa' name='GrouperSysAdmin'
gsh 4% RuleApi.groupIntersection(subjectActAs, itStaff_includes, employeeIncludingUphs);
edu.internet2.middleware.grouper.attr.assign.AttributeAssign: AttributeAssign
[id=b3849718eab34496a162165f29ba6b92,action=assign,attributeDefName=penn:etc:attribute:rules:rule,
  group=Group[name=penn:community:employee:itStaff_includes,uuid=59cd11b40d49446099e0409f755d9679]]
```