

Grouper rules pattern - Veto if too many members

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

[Grouper rules](#)

This is available in Grouper 2.5.23+

If a group has a membership limit, then veto additions to the group. You might want to configure a subject source so that group members do not count (since they won't be provisioned anyways). You can link this rule to a different group. e.g. if a group has two groups as members, then add this rule to all three groups, and point the count to happen at the parent group, and you will not be able to add members to child groups because the parent group is full.

Add this rule to the group where the membership is added.

Configure rule for v5+

The screenshot shows the Grouper web interface for configuring a rule for group 'test1'. The breadcrumb trail is 'Home > Root > test > test1'. The group name 'test1' is displayed with a 'Group actions' dropdown. Below the group name are tabs for 'Members', 'Privileges', and 'More'. The 'Group rules' section has a 'Rules actions' dropdown. Under 'Documentation', the 'Rule pattern' is set to 'Veto if too many members' (marked with a red star). Below this, the 'Maximum number of members allowed' is set to '10' (also marked with a red star). A note states: 'Enter 10 if the 11th member should be vetoed.' At the bottom are 'Save' and 'Cancel' buttons.

Configure rule for v4 and previous

Java example

```

GrouperSession grouperSession = GrouperSession.startRootSession();
Group maxGroup = new GroupSave(grouperSession).assignName("stem:maxGroup").assignCreateParentStemsIfNotExist(
true).save();
Group memberGroup = new GroupSave(grouperSession).assignName("stem:memberGroup").
assignCreateParentStemsIfNotExist(true).save();

maxGroup.addMember(memberGroup.toSubject());

//add rule on a group that checks a membership count and vetos if at limit already
AttributeAssign attributeAssign = memberGroup
    .getAttributeDelegate().addAttribute(RuleUtils.ruleAttributeDefName()).getAttributeAssign();

AttributeValueDelegate attributeValueDelegate = attributeAssign.getAttributeValueDelegate();

attributeValueDelegate.assignValue(
    RuleUtils.ruleActAsSubjectSourceIdName(), "g:isa");
attributeValueDelegate.assignValue(
    RuleUtils.ruleActAsSubjectIdName(), "GrouperSystem");

//subject use means membership add, privilege assign, permission assign, etc.
attributeValueDelegate.assignValue(
    RuleUtils.ruleCheckTypeName(), RuleCheckType.membershipAdd.name());

attributeValueDelegate.assignValue(
    RuleUtils.ruleIfConditionEnumName(), RuleIfConditionEnum.groupHasTooManyMembers.name());

// if you are not checking the group with the rule on it, identify that group to check here
attributeValueDelegate.assignValue(
    RuleUtils.ruleIfOwnerNameName(), maxGroup.getName());

attributeValueDelegate.assignValue(
    RuleUtils.ruleIfConditionEnumArg0Name(), "1");

// if checking subject sources, put them here comma separated (the sourceIds)
attributeValueDelegate.assignValue(
    RuleUtils.ruleIfConditionEnumArg1Name(), "jdbc");

attributeValueDelegate.assignValue(
    RuleUtils.ruleThenEnumName(), RuleThenEnum.veto.name());

//key which would be used in UI messages file if applicable
attributeValueDelegate.assignValue(
    RuleUtils.ruleThenEnumArg0Name(), "rule.group.has.too.many.members");

//error message (if key in UI messages file not there)
attributeValueDelegate.assignValue(
    RuleUtils.ruleThenEnumArg1Name(), "Group has too many members");

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

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

```

GSH shorthand method

```
// get a Subject object for the shorthand method
actAsSubject = findSubject("grouperSystem") ; // or any subjectIdentifier you want to use

// get a group object for the shorthand method ( can also be used for the memberGroup object if needed )
maxGroup = GroupFinder.findByName("groupname", true)

// check one group for member count
RuleApi.vetoMembershipIfTooManyMembers(actAsSubject, maxGroup, null, 1, null, "rule.group.has.too.many.
members", "Group has too many members");

// check another group for member count
RuleApi.vetoMembershipIfTooManyMembers(actAsSubject, memberGroup, maxGroup, 1, "jdbc", "rule.group.has.too.many.
members", "Group has too many members");
```