

# Grouper 2.2.0 vs 2.1.5 Performance

The following is performance results from Grouper API calls using both Grouper 2.2.0 and Grouper 2.1.5. Each call was made several times and an average was taken. Both Grouper instances were prepopulated with equivalent data before performing this test. Default configuration was used with the exception of database settings in grouper.hibernate.properties. Also, the following was added to the grouper.properties file for the 2.2 config to match the 2.1 settings.

```
groups.create.grant.all.read = true
groups.create.grant.all.view = true
```

Here's a summary of what was prepopulated:

Groups: 126,801  
Stems: 105,921  
Immediate memberships and privileges: 1,074,446  
Attribute assignments: 125,400  
Attribute def names: 41,895  
Permissions: 585,200

This data was mostly added by running edu.internet2.middleware.grouper.helper.LoadData.

Note that these tests were run on a test database server that's shared with other applications so your results will vary... A single Oracle 11g database instance was used for both Grouper versions. Both versions were installed on separate schemas. The test was executed by:

1. Using the Grouper API to add any required data for the operation. For instance, the effective membership add/delete tests required creating a group with 1000 members.
2. Then using the Grouper API to run the operation several times (mostly either 10 times or 100 times) and taking an average of the time.

Operation	Grouper 2.2.0 (ms)	Grouper 2.1.5 (ms)
GroupFinder.findByName()	2	2
Group.hasImmediateMember(Subject)	19	21
Group.hasEffectiveMember(Subject)	15	18
Group.hasMember(Subject)	15	14
Group.hasOptin(Subject)	15	19
Group.getPrivs(Subject)	15	17
Group.getUpdaters()	2	2
Group.getEffectiveMembers()	2	2
Group.getEffectiveMemberships()	3	3
Group.getImmediateMembers()	2	2
Group.getImmediateMemberships()	3	3
Group.getMembers()	2	2
Group.getMemberships()	3	4
MemberFinder.findBySubject()	14	52
Member.getEffectiveMemberships()	4	5
Member.getImmediateMemberships()	4	5
Member.getMemberships()	2	2
StemFinder.findByName()	1	1
Stem.getPrivs(Subject)	25	20
Stem.hasCreate(Subject)	13	15
Stem.getStemmers()	2	2
Group create	132	124
Group delete	84	81
Role create	133	132
Attribute def name create	25	22
Assign role permission	30	39

Remove role permission	20	21
Attribute def name delete	19	18
Role delete	82	78
Stem create	106	53
Stem delete	46	41
AttributeDef create (type=perm)	69	52
AttributeDef delete (type=perm)	66	66
Membership add	32	33
Membership delete	17	17
Membership add where member is a group	43	39
Membership delete where member is a group	22	21
Membership add causes composite	38	28
Membership delete causes composite	26	35
Group privilege (update) add	13	15
Group privilege (update) delete	12	12
Stem privilege (create) add	13	14
Stem privilege (create) delete	11	11
Effective membership (1000) add	20	16
Effective membership (1000) delete	14	13

