

# Grouper provisioning framework logging

<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>
---------------------------	---	--------------------------------	--	---	--

## When asking for support

Turn on "Log all objects verbose" to DB under advanced in provisioning configuration. You can optionally also enter in a group name and subject id to focus on.

Do a provisioning run that shows an error

Go to the daemon log for that run, scroll to the right, and copy the full output

Note, if you are doing a large run you might want to turn off command logging

Export the grouper-loader properties file from UI. Take the section for that provisioner (no need to send configs for daemon). Send that along too.

Report which version your Grouper is.

## Log all objects verbose

Edit the provisioner (in advanced section at bottom) and turn on "log all objects verbose". This will log information and data (up to 10 objects) at each point in the provisioning workflow. The logs can be read from the daemon logs from the UI. Generally if you are not troubleshooting, you can leave this off.

Advanced

Less common settings

Show advanced

☐ EL?

☐ Default value (False) ☒ True ☐ False

Show advanced options. Default value is 'false'.

Thread pool size

☐ EL?

Thread pool size, will use threads for as much work as possible. Set to 1 to not use threads. Default value is '5'.

Can full sync

☐ EL?

☒ Default value (True) ☐ True ☐ False

Can full sync. Default value is 'true'.

Recalculate all real time events

☐ EL?

☒ Default value (False) ☐ True ☐ False

If the target should be checked before sending actions. e.g. if an addMember is made to a provisionable group, the check the target to see if the entity is already a member first. Default value is 'false'.

Debug log

☐ EL?

☒ Default value (False) ☐ True ☐ False

enable debug log. Default value is 'false'.

Log all objects verbose

☐ EL?

☐ Default value (False) ☒ True ☐ False

Log each step in provision workflow (to debug issues). Default value is 'false'.

Log all objects verbose to log file

☐ EL?

☒ Default value (True) ☐ True ☐ False

If this log should be logged to log file. Default value is 'true'.

Log all objects verbose to daemon DB log

☐ EL?

☒ Default value (True) ☐ True ☐ False

If this log should be logged to DB log (accessible from UI). Default value is 'true'.

Log all objects for these subject IDs

☐ EL?

Log information about these subject IDs (comma separated less than 10). Blank for random ten. Note, it will still log other objects up to ten, but will include these.

Log all objects for these group names

☐ EL?

Log information about these group names (comma separated less than 10). Note this is the UI id path. e.g. a.b:c  
Blank for random ten. Note, it will still log other objects up to ten, but will include these.

## Filter based on group(s) and/or subject(s)

In 4.2.1 you can enter several subject IDs or group names in the config so that those objects are logged, if you are troubleshooting a certain group and/or subject. Note, as the provisioner loops through data, it will log other objects too, but the ones specified will be included. (see config above)

## Log level

Set this in log4j.properties (optional and not commonly set). If you don't have this as DEBUG, then it will debug as ERROR

```
log4j.logger.edu.internet2.middleware.grouper.app.provisioning.GrouperProvisioningObjectLog = DEBUG
```

## Analyze logs

After retrieve from grouper and target

```
2020-09-06 19:12:54,186: [main] DEBUG GrouperProvisioningObjectLog.debug(51) - - Provisioner 'sqlProvTest'
object log after state: 'retrieveAllData': {finalLog=false, state=retrieveAllData,
provisionerClass=SqlMembershipProvisioner, configId=sqlProvTest, provisioningType=fullProvisionFull,
retrieveTargetGroupsMillis=2, targetGroupCount=2, retrieveTargetMshipsMillis=0, targetMshipCount=0,
retrieveSyncGroupsMillis=4, syncGroupCount=2, retrieveSyncEntitiesMillis=0, syncEntityCount=4,
retrieveSyncMshipsMillis=2, syncMshipCount=4, retrieveGrouperGroupsMillis=7, grouperGroupCount=3,
retrieveGrouperEntitiesMillis=3, grouperEntityCount=6, retrieveGrouperMshipsMillis=4, grouperMshipCount=6,
fixSyncObjectStoreCount=11, retrieveAllDataMillis=25}
Grouper provisioning groups (3):
1. Group(id: 958274a44b394fdcaf06bfbdb66c05dc4, idIndex: 10009, name: test3:testGroup3, displayName: test3:
testGroup3)
2. Group(id: a596f943aa214f7a93f9c3e96c9b910b, idIndex: 10020, name: test4:testGroup4, displayName: test4:
testGroup4)
3. Group(id: d41c08b1bda04ad3b71fcd62d30d21a, idIndex: 10007, name: test:testGroup, displayName: test:
testGroup)
Grouper provisioning entities (6):
1. Entity(id: 882490d3b12a4c299a8b9eac6cb10529, name: my name is test.subject.0, attr[description]: description.
test.subject.0, attr[subjectId]: test.subject.0, attr[subjectIdentifier0]: id.test.subject.0)
2. Entity(id: 3aa0710792f4422e965a07cfd2e68bbb, name: my name is test.subject.4, attr[description]: description.
test.subject.4, attr[subjectId]: test.subject.4, attr[subjectIdentifier0]: id.test.subject.4)
3. Entity(id: 48cdc3040a7542dbbbb8271164c1ba98, name: my name is test.subject.5, attr[description]: description.
test.subject.5, attr[subjectId]: test.subject.5, attr[subjectIdentifier0]: id.test.subject.5)
4. Entity(id: b995e47c3bea43a9a78e6e3d776194fd, name: my name is test.subject.6, attr[description]: description.
test.subject.6, attr[subjectId]: test.subject.6, attr[subjectIdentifier0]: id.test.subject.6)
5. Entity(id: 03a4db85778249bdbc6ee5915d7319, name: my name is test.subject.7, attr[description]: description.
test.subject.7, attr[subjectId]: test.subject.7, attr[subjectIdentifier0]: id.test.subject.7)
6. Entity(id: bf3a8cab7949471d9d3f6c5437835dc3, name: my name is test.subject.8, attr[description]: description.
test.subject.8, attr[subjectId]: test.subject.8, attr[subjectIdentifier0]: id.test.subject.8)
Grouper provisioning memberships (6):
1. Mship(group: testGroup3, entity: my name is test.subject.5, groupId: 958274a44b394fdcaf06bfbdb66c05dc4,
entityId: 48cdc3040a7542dbbbb8271164c1ba98, id: 13bfa36809c441d8b7556da9309bc458:
5ac5814fa7b94239a2db14e3f6c28e5e)
2. Mship(group: testGroup3, entity: my name is test.subject.4, groupId: 958274a44b394fdcaf06bfbdb66c05dc4,
entityId: 3aa0710792f4422e965a07cfd2e68bbb, id: 9268ba811192f41fba75329f3bd3136a4:
5ac5814fa7b94239a2db14e3f6c28e5e)
3. Mship(group: testGroup4, entity: my name is test.subject.6, groupId: a596f943aa214f7a93f9c3e96c9b910b,
entityId: b995e47c3bea43a9a78e6e3d776194fd, id: 1cf7b798ad7b4940b3f868153a788b71:
69a7cdefbf554af283d8a31c0f3e6048)
4. Mship(group: testGroup4, entity: my name is test.subject.7, groupId: a596f943aa214f7a93f9c3e96c9b910b,
entityId: 03a4db85778249bdbc6ee5915d7319, id: 346585464b2742b6b2227e10cb2fe2a0:
69a7cdefbf554af283d8a31c0f3e6048)
5. Mship(group: testGroup, entity: my name is test.subject.8, groupId: d41c08b1bda04ad3b71fcd62d30d21a,
entityId: bf3a8cab7949471d9d3f6c5437835dc3, id: 75cc77a4e919402d9b376047185353b1:
48f2847d70ab4a8e8d11e7e546c66905)
6. Mship(group: testGroup, entity: my name is test.subject.0, groupId: d41c08b1bda04ad3b71fcd62d30d21a,
entityId: 882490d3b12a4c299a8b9eac6cb10529, id: e788ddf16e4a4db3906f5700ace7d2ec:
48f2847d70ab4a8e8d11e7e546c66905)
Target provisioning groups (2):
1. Group(id: test3:testGroup3, attr[groupName]: HashSet (1): [0]: test3:testGroup3, attr[subjectId]: HashSet
(2): [0]: test.subject.5, [1]: test.subject.4)
2. Group(id: test:testGroup, attr[groupName]: HashSet (1): [0]: test:testGroup, attr[subjectId]: HashSet (2):
[0]: test.subject.1, [1]: test.subject.0)
Target provisioning entities (0)
Target provisioning memberships (0)
```

After translate grouper to target objects

```

2020-09-06 19:12:54,186: [main] DEBUG GrouperProvisioningObjectLog.debug(51) - - Provisioner 'sqlProvTest'
object log after state: 'translateGrouperToTarget': {finalLog=false, state=translateGrouperToTarget,
provisionerClass=SqlMembershipProvisioner, configId=sqlProvTest, provisioningType=fullProvisionFull,
retrieveTargetGroupsMillis=2, targetGroupCount=2, retrieveTargetMshipsMillis=0, targetMshipCount=0,
retrieveSyncGroupsMillis=4, syncGroupCount=2, retrieveSyncEntitiesMillis=0, syncEntityCount=4,
retrieveSyncMshipsMillis=2, syncMshipCount=4, retrieveGrouperGroupsMillis=7, grouperGroupCount=3,
retrieveGrouperEntitiesMillis=3, grouperEntityCount=6, retrieveGrouperMshipsMillis=4, grouperMshipCount=6,
fixSyncObjectStoreCount=11, retrieveAllDataMillis=25}
Grouper target groups (3):
1. Group(attr[groupName]: test3:testGroup3, attr[subjectId]: HashSet (2): [0]: test.subject.5, [1]: test.
subject.4)
2. Group(attr[groupName]: test4:testGroup4, attr[subjectId]: HashSet (2): [0]: test.subject.7, [1]: test.
subject.6)
3. Group(attr[groupName]: test:testGroup, attr[subjectId]: HashSet (2): [0]: test.subject.8, [1]: test.subject.
0)
Grouper target entities (0)
Grouper target memberships (0)

```

After translate matching id's

```

2020-09-06 19:12:54,187: [main] DEBUG GrouperProvisioningObjectLog.debug(51) - - Provisioner 'sqlProvTest'
object log after state: 'idTargetObjects': {finalLog=false, state=idTargetObjects,
provisionerClass=SqlMembershipProvisioner, configId=sqlProvTest, provisioningType=fullProvisionFull,
retrieveTargetGroupsMillis=2, targetGroupCount=2, retrieveTargetMshipsMillis=0, targetMshipCount=0,
retrieveSyncGroupsMillis=4, syncGroupCount=2, retrieveSyncEntitiesMillis=0, syncEntityCount=4,
retrieveSyncMshipsMillis=2, syncMshipCount=4, retrieveGrouperGroupsMillis=7, grouperGroupCount=3,
retrieveGrouperEntitiesMillis=3, grouperEntityCount=6, retrieveGrouperMshipsMillis=4, grouperMshipCount=6,
fixSyncObjectStoreCount=11, retrieveAllDataMillis=25}
Grouper target groups (3):
1. Group(matchingId: test3:testGroup3, attr[groupName]: test3:testGroup3, attr[subjectId]: HashSet (2): [0]:
test.subject.5, [1]: test.subject.4)
2. Group(matchingId: test4:testGroup4, attr[groupName]: test4:testGroup4, attr[subjectId]: HashSet (2): [0]:
test.subject.7, [1]: test.subject.6)
3. Group(matchingId: test:testGroup, attr[groupName]: test:testGroup, attr[subjectId]: HashSet (2): [0]: test.
subject.8, [1]: test.subject.0)
Grouper target entities (0)
Grouper target memberships (0)
Target provisioning groups (2):
1. Group(id: test3:testGroup3, matchingId: [test3:testGroup3], attr[groupName]: HashSet (1): [0]: test3:
testGroup3, attr[subjectId]: HashSet (2): [0]: test.subject.5, [1]: test.subject.4)
2. Group(id: test:testGroup, matchingId: [test:testGroup], attr[groupName]: HashSet (1): [0]: test:testGroup,
attr[subjectId]: HashSet (2): [0]: test.subject.1, [1]: test.subject.0)
Target provisioning entities (0)
Target provisioning memberships (0)

```

After compare

```
2020-09-06 19:12:54,187: [main] DEBUG GrouperProvisioningObjectLog.debug(51) - - Provisioner 'sqlProvTest'
object log after state: 'compareTargetObjects': {finalLog=false, state=compareTargetObjects,
provisionerClass=SqlMembershipProvisioner, configId=sqlProvTest, provisioningType=fullProvisionFull,
retrieveTargetGroupsMillis=2, targetGroupCount=2, retrieveTargetMshipsMillis=0, targetMshipCount=0,
retrieveSyncGroupsMillis=4, syncGroupCount=2, retrieveSyncEntitiesMillis=0, syncEntityCount=4,
retrieveSyncMshipsMillis=2, syncMshipCount=4, retrieveGrouperGroupsMillis=7, grouperGroupCount=3,
retrieveGrouperEntitiesMillis=3, grouperEntityCount=6, retrieveGrouperMshipsMillis=4, grouperMshipCount=6,
fixSyncObjectStoreCount=11, retrieveAllDataMillis=25, groupInsertsAfterCompare=3, groupDeletesAfterCompare=2}
Target inserts groups (3):
1. Group(matchingId: test4:testGroup4, attr[groupName]: test4:testGroup4, attr[subjectId]: HashSet (2): [0]:
test.subject.7, [1]: test.subject.6, attributes_to_insert: 3)
2. Group(matchingId: test3:testGroup3, attr[groupName]: test3:testGroup3, attr[subjectId]: HashSet (2): [0]:
test.subject.5, [1]: test.subject.4, attributes_to_insert: 3)
3. Group(matchingId: test:testGroup, attr[groupName]: test:testGroup, attr[subjectId]: HashSet (2): [0]: test.
subject.8, [1]: test.subject.0, attributes_to_insert: 3)
Target inserts entities (0)
Target inserts memberships (0)
Target updates groups (0)
Target updates entities (0)
Target updates memberships (0)
Target deletes groups (2):
1. Group(id: test3:testGroup3, matchingId: [test3:testGroup3], attr[groupName]: HashSet (1): [0]: test3:
testGroup3, attr[subjectId]: HashSet (2): [0]: test.subject.5, [1]: test.subject.4)
2. Group(id: test:testGroup, matchingId: [test:testGroup], attr[groupName]: HashSet (1): [0]: test:testGroup,
attr[subjectId]: HashSet (2): [0]: test.subject.1, [1]: test.subject.0)
Target deletes entities (0)
Target deletes memberships (0)
```

At end

```
2020-09-06 19:12:54,192: [main] DEBUG GrouperProvisioningObjectLog.debug(51) - - Provisioner 'sqlProvTest'
object log after state: 'sendChangesToTarget': {finalLog=false, state=sendChangesToTarget,
provisionerClass=SqlMembershipProvisioner, configId=sqlProvTest, provisioningType=fullProvisionFull,
retrieveTargetGroupsMillis=2, targetGroupCount=2, retrieveTargetMshipsMillis=0, targetMshipCount=0,
retrieveSyncGroupsMillis=4, syncGroupCount=2, retrieveSyncEntitiesMillis=0, syncEntityCount=4,
retrieveSyncMshipsMillis=2, syncMshipCount=4, retrieveGrouperGroupsMillis=7, grouperGroupCount=3,
retrieveGrouperEntitiesMillis=3, grouperEntityCount=6, retrieveGrouperMshipsMillis=4, grouperMshipCount=6,
fixSyncObjectStoreCount=11, retrieveAllDataMillis=25, groupInsertsAfterCompare=3, groupDeletesAfterCompare=2}
```