

Grouper changes v2.0

This document lists instructions for people with existing groups installations on how to upgrade to newer versions of grouper (or grouper related products). If you notice something missing please let us know. The instructions are in descending order based on date/release. You will find instructions below for Grouper, Grouper-ws, Grouper-ui, etc. It is assumed if you are running grouper-ui that you will perform both the grouper upgrade notes, and the grouper-ui upgrade notes. It is understood that you will get the new source/javadoc/etc files, this document addresses configurations, jars, etc. Note that for major upgrades, you should [follow the upgrade steps](#). For minor upgrades, that instructions should be sufficient.

Grouper

- v2.0.2: compare the grouper.hibernate.properties with grouper.hibernate and re-arrange the driver/dialect, and optionally blank them out so Grouper can auto-detect. You can also do this in the grouper-loader.properties.
- v2.0.2: compare sources.xml with sources.example.xml and you should add an inclause to any jdbc (not jdbc2) sources:

```

    <!-- if you are going to use the inclause attribute
         on the search to make the queries batchable when searching
         by id or identifier -->
    <init-param>
      <param-name>useInClauseForIdAndIdentifier</param-name>
      <param-value>true</param-value>
    </init-param>

    <!-- comma separate the identifiers for this row, this is for the findByIdentifiers if using an in clause
-->
    <init-param>
      <param-name>identifierAttributes</param-name>
      <param-value>LOGINID</param-value>
    </init-param>

<search>
  <searchType>searchSubject</searchType>
  <param>
    <param-name>sql</param-name>
    <param-value>
select
  s.subjectid as id, s.name as name,
  (select sa2.value from subjectattribute sa2 where name='name' and sa2.SUBJECTID = s.subjectid) as lfname,
  (select sa3.value from subjectattribute sa3 where name='loginid' and sa3.SUBJECTID = s.subjectid) as
loginid,
  (select sa4.value from subjectattribute sa4 where name='description' and sa4.SUBJECTID = s.subjectid) as
description,
  (select sa5.value from subjectattribute sa5 where name='email' and sa5.SUBJECTID = s.subjectid) as email
from
  subject s
where
  {inclause}
    </param-value>
  </param>
  <param>
    <param-name>inclause</param-name>
    <param-value>
s.subjectid = ?
    </param-value>
  </param>
</search>
<search>
  <searchType>searchSubjectByIdentifier</searchType>
  <param>
    <param-name>sql</param-name>
    <param-value>
select
  s.subjectid as id, s.name as name,
  (select sa2.value from subjectattribute sa2 where name='name' and sa2.SUBJECTID = s.subjectid) as lfname,
  (select sa3.value from subjectattribute sa3 where name='loginid' and sa3.SUBJECTID = s.subjectid) as
loginid,
  (select sa4.value from subjectattribute sa4 where name='description' and sa4.SUBJECTID = s.subjectid) as
description,
  (select sa5.value from subjectattribute sa5 where name='email' and sa5.SUBJECTID = s.subjectid) as email
from
  subject s, subjectattribute a
where
  a.name='loginid' and s.subjectid = a.subjectid and {inclause}
    </param-value>
  </param>
  <param>
    <param-name>inclause</param-name>
    <param-value>
a.value = ?
    </param-value>
  </param>
</search>

```

- v2.0.2: compare sources.xml with sources.example.xml and you should probably add a max page size to each source:

```
<!-- on a findPage() this is the most results returned -->
<init-param>
  <param-name>maxPageSize</param-name>
  <param-value>100</param-value>
</init-param>
```

- v2.0.2: compare grouper.properties with grouper.example.properties, and add this section to the grouper.properties

```
#####
## Subject settings
#####

# if finding across multiple threadable sources, use threads to do the work faster
subjects.allPage.useThreadForkJoin = false

# if finding across multiple threadable sources, use threads to do the work faster
subjects.idOrIdentifier.useThreadForkJoin = false

# if the creator and last updater should be group subject attributes (you get
# a performance gain if you set to false, but if true you can see subject id from UI in 2.0
subjects.group.useCreatorAndModifierAsSubjectAttributes = true

# if we should use a root session if one isnt started for subject lookups (behavior in v2.0-
subjects.startRootSessionIfOneIsntStarted = true
```

- v2.0.2: merge grouper.hibernate.properties with grouper.hibernate.example.properties

```
OLD:
hibernate.cache.provider_class = org.hibernate.cache.EhCacheProvider
```

```
NEW:
hibernate.cache.region.factory_class = net.sf.ehcache.hibernate.EhCacheRegionFactory
```

- v2.0.2: The default sort string for groups was changed from "name" to "displayExtension". Note that the admin UI shows displayExtension when showing membership lists, but the lite UI shows the displayName. So even with the new default, the membership list in the lite UI will not look like it is sorted correctly. If your users use the lite UI and not the admin UI, you may want to use "displayName" instead for now. Perhaps in 2.2, the lite UI will show the displayExtension instead. If you change the sort string, you should sync the member attributes. See [Member search and sort columns](#). Here's the new default in sources.xml:

```

<source adapterClass="edu.internet2.middleware.grouper.GrouperSourceAdapter">
  <id>g:gsa</id>
  <name>Grouper: Group Source Adapter</name>
  <type>group</type>

  <init-param>
    <param-name>subjectVirtualAttribute_0_searchAttribute0</param-name>
    <param-value>${subject.getAttributeValue('name')},${subject.getAttributeValue(
('displayName')},${subject.getAttributeValue('alternateName'))}</param-value>
  </init-param>
  <init-param>
    <param-name>sortAttribute0</param-name>
    <param-value>displayExtension</param-value>
  </init-param>
  <init-param>
    <param-name>searchAttribute0</param-name>
    <param-value>searchAttribute0</param-value>
  </init-param>
  <!-- on a findPage() this is the most results returned -->
  <init-param>
    <param-name>maxPageSize</param-name>
    <param-value>100</param-value>
  </init-param>
  <internal-attribute>searchAttribute0</internal-attribute>
</source>

```

- v2.0.1: Note that grouper-loader.properties has loader.autoadd.typesAttributes default to true [GRP-672](#)
- v2.0.1: Merge ehcache.xml with ehcache.example.xml. Unless you previously customized this configuration file, you should be able to just copy the new configuration. The change here was to add caching for Hib3PITFieldDAO.findById() and PITField. And also to no longer expire UpdateTimestampsCache and set the default cache to expire after 1 second rather than to never expire. Here are the cache settings for the updated caches:

```

<defaultCache
  maxElementsInMemory="1000"
  eternal="false"
  timeToIdleSeconds="1"
  timeToLiveSeconds="1"
  overflowToDisk="false"
/>

<cache name="net.sf.hibernate.cache.UpdateTimestampsCache"
  maxElementsInMemory="5000"
  eternal="true"
  overflowToDisk="true"
/>

<cache name="org.hibernate.cache.UpdateTimestampsCache"
  maxElementsInMemory="5000"
  eternal="true"
  overflowToDisk="true"
/>

<cache name="edu.internet2.middleware.grouper.pit.PITField"
  maxElementsInMemory="1000"
  eternal="false"
  timeToIdleSeconds="30"
  timeToLiveSeconds="120"
  overflowToDisk="false"
/>

<cache name="edu.internet2.middleware.grouper.internal.dao.hib3.Hib3PITFieldDAO.FindById"
  maxElementsInMemory="1000"
  eternal="false"
  timeToIdleSeconds="30"
  timeToLiveSeconds="120"
  overflowToDisk="false"
/>

```

- v2.0.0: Merge grouper.properties with grouper.example.properties
 - Note, this should be true, and set where you want the system attributes to live:

```

# root stem in grouper where built in attributes are put
grouper.attribute.rootStem = etc:attribute

# if the attribute loader attributes, and other attributes should be autoconfigured (created, etc)
grouper.attribute.loader.autoconfigure = true

```

- The following general settings have been added.

```

#put the URL which will be used e.g. in emails to users. include the webappName at the end, and
nothing after that.
#e.g. https://server.school.edu/grouper/
grouper.ui.url =

```

- Subject API changed to support [sort and search strings for subjects](#). You can use the sources.xml file to map subject attributes to sort and search fields that are stored in Grouper. Every source must be configured for at least one sort string and one search string. Most of the sources are configured in sources.xml but the internal and external sources are configured in grouper.properties. Here are the configuration updates for grouper.properties.

```

# Search and sort strings for internal users
internalSubjects.searchAttribute0.el = ${subject.name},${subject.id}
internalSubjects.sortAttribute0.el = ${subject.name}

...

# By default, all users have access to sort using any of the sort strings in the member table and
search using any of the search strings in the member table.
# You can restrict to wheel only or to a certain group.
#security.member.sort.string0.allowOnlyGroup = etc:someGroup
#security.member.sort.string1.allowOnlyGroup = etc:someGroup
#security.member.sort.string2.wheelOnly = true
#security.member.sort.string3.wheelOnly = true
#security.member.sort.string4.wheelOnly = true
#security.member.search.string0.allowOnlyGroup = etc:someGroup
#security.member.search.string1.allowOnlyGroup = etc:someGroup
#security.member.search.string2.wheelOnly = true
#security.member.search.string3.wheelOnly = true
#security.member.search.string4.wheelOnly = true

#####
## Member sort and search
#####

# Attributes of members are kept in the grouper_members table to allow easy sorting and searching
(for instance when listing group members).
# When performing a sort or search and an index is not specified, then a default index will be
used as configured below. The value is comma-separated,
# so that if the user does not have access to the first index, then next will be tried and so
forth.
# Note: all sources should have attributes configured for all default indexes.
member.search.defaultIndexOrder=0
member.sort.defaultIndexOrder=0

```

- There is additional support for Grouper hooks.

```

#implement an attribute def hook by extending edu.internet2.middleware.grouper.hooks.
AttributeDefHooks
#hooks.attributeDef.class=edu.yourSchool.it.YourSchoolAttributeDefHooks,edu.yourSchool.it.
YourSchoolAttributeDefHooks2

#implement an attribute def name hook by extending edu.internet2.middleware.grouper.hooks.
AttributeDefNameHooks
#hooks.attributeDefName.class=edu.yourSchool.it.YourSchoolAttributeDefNameHooks,edu.yourSchool.it.
YourSchoolAttributeDefNameHooks2

#implement an attribute assign hook by extending edu.internet2.middleware.grouper.hooks.
AttributeAssignHooks
#hooks.attributeAssign.class=edu.yourSchool.it.YourSchoolAttributeAssignHooks,edu.yourSchool.it.
YourSchoolAttributeAssignHooks2

#implement an attribute assign hook by extending edu.internet2.middleware.grouper.hooks.
AttributeAssignValueHooks
#hooks.attributeAssignValue.class=edu.yourSchool.it.YourSchoolAttributeAssignValueHooks,edu.
yourSchool.it.YourSchoolAttributeAssignValueHooks2

#implement an external subject hook by extending edu.internet2.middleware.grouper.hooks.
ExternalSubjectHooks
#hooks.externalSubject.class=edu.yourSchool.it.YourSchoolExternalSubjectHooks

```

- Grouper rules are new in 2.0.

```
#####
## Rules
#####

# Rules users who are in the following group can use the actAs field to act as someone else
# You can put multiple groups separated by commas. e.g. a:b:c, e:f:g
# You can put a single entry as the group the calling user has to be in, and the grouper the actAs
has to be in
# separated by 4 colons
# e.g. if the configured values is: a:b:c, e:f:d :::: r:e:w, x:e:w
# then if the calling user is in a:b:c or x:e:w, then the actAs can be anyone
# if not, then if the calling user is in e:f:d, then the actAs must be in r:e:w. If multiple
rules, then
# if one passes, then it is a success, if they all fail, then fail.
rules.act.as.group =

# any actAs subject in this group has access to more objects when the EL fires on
# the IF or THEN EL clause
rules.accessToApiInEl.group =

# cache the decision to allow a user to actAs another, so it doesnt have to be calculated each time
# defaults to 30 minutes
rules.act.as.cache.minutes = 30

# uuids (comma separated) of the attribute assign record which is the rule type to the owner object
# e.g. SELECT gaagv.attribute_assign_id FROM grouper_attr_asn_group_v gaagv WHERE gaagv.
attribute_def_name_name LIKE ':%rule' AND gaagv.group_name = 'stem:a'
# make sure log info level is set for RuleEngine
# log4j.logger.edu.internet2.middleware.grouper.rules.RuleEngine = INFO
rules.attributeAssignTypeIdsToLog = abc1234abc123, def456def345

# if this is true, then log a lot of info about why rules do or do not fire... only turn on
temporarily
# since it takes a lot of resources... note you need log DEBUG set for the rules engine in log4j.
properties too e.g.
# log4j.logger.edu.internet2.middleware.grouper.rules = DEBUG
rules.logWhyRulesDontFire = false

# put in fully qualified classes to add to the EL context. Note that they need a default
constructor
# comma separated. The alias will be the simple class name without a first cap.
# e.g. if the class is test.Test the alias is "test"
rules.customElClasses =

# If the CHECK, IF, and THEN are all exactly what is needed for managing inherited stem privileges
# Then allow an actAs GrouperSystem in source g:isa
rules.allowActAsGrouperSystemForInheritedStemPrivileges =

# If not blank, then keep email templates in this folder instead of classpath
# If in classpath, it is classpath: grouperRulesEmailTemplates/someTemplate.txt
rules.emailTemplatesFolder =
```

- If you want to keep track of the last immediate membership update of a group.

```
# If true, when an immediate membership changes for a group (either a privilege or a list member),
# then an update will be made to the lastImmediateMembershipChange property for the group.
groups.updateLastImmediateMembershipTime = true
```

- Additional email settings

```
#leave blank or false for no ssl, true for ssl
#mail.smtp.ssl =

#leave blank for default (probably 25), if ssl is true, default is 465, else specify
#mail.smtp.port =

#when running junit tests, this is the address that will be used
#mail.test.address = a@b.c
```

- Additional junit settings

```
# if the external subject tests should be included when running all tests, note you need the
jabber attribute in the view (default false)
junit.test.externalSubjects = false

# if the group sync should be tested... note you need the demo server available to test this, or
change some settings...
junit.test.groupSync = false
junit.test.groupSync.url = https://grouperdemo.internet2.edu/grouper-ws_v2_0_0/servicesRest
junit.test.groupSync.user = remoteUser
junit.test.groupSync.password = R:/pass/grouperDemoRemoteUser.pass
#folder where the user can create/stem which the user can use to run tests
junit.test.groupSync.folder = test2:whateverFolder
#this is true unless testing to an older grouper which doesnt support this
junit.test.groupSync.pushAddExternalSubjectIfNotExist = true
junit.test.groupSync.createRemoteFolderIfNotExist = true
junit.test.groupSync.remoteSourceId = grouperExternal
junit.test.groupSync.remoteReadSubjectId = identifier
junit.test.groupSync.remoteWriteSubjectId = identifier
```

- Grouper Permission Limits are new in 2.0.

```
#####
## centrally managed permissions
#####

# if the permissions limits should be readable and updatable by GrouperAll (set when created)...
grouper.permissions.limits.builtin.createAs.public = true

# if the permissions limits should be readable and updatable by GrouperAll (set when created)...
grouper.permissions.limits.builtin.displayExtension.limitAmountLessThan = amount less than
grouper.permissions.limits.builtin.displayExtension.limitAmountLessThanOrEqual = amount less than
or equal to
grouper.permissions.limits.builtin.displayExtension.limitExpression = Expression
grouper.permissions.limits.builtin.displayExtension.limitIpOnNetworkRealm = ipAddress on network
realm
grouper.permissions.limits.builtin.displayExtension.limitIpOnNetworks = ipAddress on networks
grouper.permissions.limits.builtin.displayExtension.limitLabelsContain = labels contains
grouper.permissions.limits.builtin.displayExtension.limitWeekday9to5 = Weekday 9 to 5

# el classes to add to the el context for a limitExpression. Comma-separated fully qualified
classnames
grouper.permissions.limits.el.classes =

# permission limits linked to subclasses of edu.internet2.middleware.grouper.permissions.limits.
PermissionLimitBase
#grouper.permissions.limits.logic.someName.limitName =
#grouper.permissions.limits.logic.someName.logicClass =

# if you are doing ip address limits, you can put realms here
# grouper.permissions.limits.realm.someName = 1.2.3.4/24, 2.3.4.5/16
```

- External subjects are new in 2.0


```
#####
## External subjects
#####

#manages the description of a user automatically
externalSubjects.desc.el = [unverifiedInfo] ${grouperUtil.appendIfNotBlankString(externalSubject.
name, ' - ', externalSubject.institution)} [externalUserID] ${externalSubject.identifier}

#search and sort strings added to member objects
externalSubjects.searchAttribute0.el = ${subject.name},${subjectUtils.defaultIfBlank(subject.
getAttributeValue("institution"), "")},${subjectUtils.defaultIfBlank(subject.getAttributeValue
("identifier"), "")},${subject.id},${subjectUtils.defaultIfBlank(subject.getAttributeValue
("email"), "")}
externalSubjects.sortAttribute0.el = ${subject.name}
externalSubjects.sortAttribute1.el = ${subjectUtils.defaultIfBlank(subject.getAttributeValue
("identifier"), "")}
externalSubjects.sortAttribute2.el = ${subjectUtils.defaultIfBlank(subject.getAttributeValue
("institution"), "")}

# false if the description should be managed via EL (config above)
externalSubjects.desc.manual = false

# quartz cron where subjects are recalculated if necessary (empty means dont run), e.g. everyday
at 3am
externalSubjects.calc.fields.cron = 0 0 3 * * ?

externalSubjects.name.required = true
externalSubjects.email.required = false
externalSubjects.email.enabled = true

# these field names (uuid, institution, identifier, uuid, email, name) or attribute names
# will be toLowered, and appended with comma separators. e.g. if you add attributes, add them
here too
externalSubjects.searchStringFields = name, institution, identifier, uuid, email

externalSubjects.institution.required = false
externalSubjects.institution.enabled = true

# note, this must be only alphanumeric lower case or underscore
# (valid db column name, subject attribute name)
#externalSubjects.attributes.jabber.systemName = jabber
#externalSubjects.attributes.jabber.required = false
# comment on column in DB (no special characters allowed)
#externalSubjects.attributes.jabber.comment = The jabber ID of the user

# if wheel or root can edit external users
externalSubjects.wheelOrRootCanEdit = true

# group which is allowed to edit external users
externalSubjects.groupAllowedForEdit =

# if the view on the external subjects should be created.
# turn this off if it doesnt compile, othrewise should be fine
externalSubjects.createView = true

#name of external subject source, defaults to grouperExternal
externalSubject.sourceId = grouperExternal
externalSubject.sourceName = External Users

# grouper can auto create a jdbc2 source for the external subjects
externalSubjects.autoCreateSource = false

# put in fully qualified classes to add to the EL context. Note that they need a default
constructor
# comma separated. The alias will be the simple class name without a first cap.
# e.g. if the class is test.Test the alias is "test"
externalSubjects.customElClasses =

# change these to affect the storage where external subjects live (e.g. to store in ldap),
# must implement each respective storable interface
```

```

externalSubjects.storage.ExternalSubjectStorable.class = edu.internet2.middleware.grouper.
externalSubjects.ExternalSubjectDbStorage
externalSubjects.storage.ExternalSubjectAttributeStorable.class = edu.internet2.middleware.grouper.
externalSubjects.ExternalSubjectAttributeDbStorage

# you can use the variables $newline$, $inviteLink$. Note, you need to change this default
message...
externalSubjectsInviteDefaultEmail = Hello,$newline$$newline$This is an invitation to register at
our site to be able to access our applications. This invitation expires in 7 days. Click on the
link below and sign in with your InCommon credentials. If you do not have InCommon credentials
you can register at a site like protectnetwork.org and use those
credentials.$newline$$newline$$inviteLink$$newline$$newline$Regards.
# default subject for email
externalSubjectsInviteDefaultEmailSubject = Register to access applications

# you can use the variables $newline$, $inviteeIdentifier$, $inviteeEmailAddress$. Note, you need
to change this default message...
externalSubjectsNotifyInviterEmail = Hello,$newline$$newline$This is a notification that user
$inviteeIdentifier$ from email address $inviteeEmailAddress$ has registered with the identity
management service. They can now use applications at this institution.$newline$$newline$Regards.
externalSubjectsNotifyInviterSubject = $inviteeIdentifier$ has registered

# number of days after which this request will expire. If -1, then will not expire
externalSubjectsInviteExpireAfterDays = 7

#put some group names comma separated for groups to auto add subjects to
externalSubjects.autoaddGroups=
#should be insert, or update, or insert,update
externalSubjects.autoaddGroupActions=insert,update
#if a number is here, expire the group assignment after a certain number of days
externalSubjects.autoaddGroupExpireAfterDays=

# add multiple group assignment actions by URL param: externalSubjectInviteName
#externalSubjects.autoadd.testingLibrary.externalSubjectInviteName=library

# comma separated groups to add for this type of invite
#externalSubjects.autoadd.testingLibrary.groups=

# should be insert, update, or insert,update
#externalSubjects.autoadd.testingLibrary.actions=insert,update

# should be insert, update, or insert,update
#externalSubjects.autoadd.testingLibrary.expireAfterDays=

#if registrations are only allowed if invited or existing...
externalSubjects.registerRequiresInvite=true

#make sure the identifier when logging in is like an email address or eppn, e.g. username@school.
edu
externalSubjects.validateIdentifierLikeEmail=true

#put regexes here, increment the 0 for multiple entries, e.g. restrict your own institution
#note, the extensions must be sequential (dont skip), regex e.g. ^.*@myschool\\.edu$
externalSubjects.regexForInvalidIdentifier.0=

```

- If your Grouper instance needs to talk to another Grouper instance (for instance to sync data):

```

#####
## Grouper client connections
## if this grouper needs to talk to another grouper, this is the client connection information
#####

# id of the source, should match the part in the property name
#grouperClient.someOtherSchool.id = someOtherSchool

# url of web service, should include everything up to the first resource to access
# e.g. https://groups.school.edu/grouperWs/servicesRest
#grouperClient.someOtherSchool.properties.grouperClient.webService.url = https://some.other.school.

```

```

edu/grouperWs/servicesRest

# login ID
#grouperClient.someOtherSchool.properties.grouperClient.webService.login = someRemoteLogin

# password for shared secret authentication to web service
# or you can put a filename with an encrypted password
#grouperClient.someOtherSchool.properties.grouperClient.webService.password = *****

# client version should match or be related to the server on the other end...
#grouperClient.someOtherSchool.properties.grouperClient.webService.client.version = v2_0_000

# this is the subject to act as local, if blank, act as GrouperSystem, specify with SubjectFinder
packed string, e.g.
# subjectIdOrIdentifier or sourceId:::subjectId or :::subjectId or sourceId:::
subjectIdentifier or :::::subjectIdentifier
# sourceId:::subjectIdOrIdentifier or :::subjectIdOrIdentifier
#grouperClient.someOtherSchool.localActAsSubject =

# the id of this source, generally the same as the name in the property name. This is mandatory
#grouperClient.someOtherSchool.source.jdbc.id = jdbc

# the part between "grouperClient.someOtherSchool.source." and ".id" links up the configs,
# in this case, "jdbc", make sure it has no special chars. sourceId can be blank if you dont want
to specify
#grouperClient.someOtherSchool.source.jdbc.local.sourceId = jdbc

# this is the identifier that goes between them, it is "id" or an attribute name. subjects
without this attribute will not be processed
#grouperClient.someOtherSchool.source.jdbc.local.read.subjectId = identifier

# this is the identifier to lookup to add a subject, should be "id" or "identifier" or
"idOrIdentifier"
#grouperClient.someOtherSchool.source.jdbc.local.write.subjectId = identifier

# sourceId of the remote system, can be blank
#grouperClient.someOtherSchool.source.jdbc.remote.sourceId = jdbc

# this is the identifier that goes between them, it is "id" or an attribute name. subjects
without this attribute will not be processed
#grouperClient.someOtherSchool.source.jdbc.remote.read.subjectId =

# this is the identifier to lookup to add a subject, should be "id" or "identifier" or
"idOrIdentifier"
#grouperClient.someOtherSchool.source.jdbc.remote.write.subjectId =

#####
## Sync to/from another grouper
## Only sync one group to one other group, do not sync one group to
## two report groups. If you need to do this, add the group to another group
#####

# we need to know where our
# connection name in grouper client connections above
#syncAnotherGrouper.testGroup0.connectionName = someOtherSchool

# incremental or push or pull or incremental_push. Note, incremental push is cron'ed and
incremental (to make sure no discrepancies arise)
#syncAnotherGrouper.testGroup0.syncType = incremental_push

# quartz cron to schedule the pull or push (incremental is automatic as events happen) (e.g. 5am
daily)
#syncAnotherGrouper.testGroup0.cron = 0 0 5 * * ?

# local group which is being synced
#syncAnotherGrouper.testGroup0.local.groupName = test:testGroup

# remote group at another grouper which is being synced

```

```
#syncAnotherGrouper.testGroup0.remote.groupName = test2:testGroup2

# if subjects are external and should be created if not exist
#syncAnotherGrouper.testGroup0.addExternalSubjectIfNotFound = true
```

- v2.0.0: Merge grouper-loader.properties with grouper-loader.example.properties

- Clear out old change log entries. Old records will be deleted from grouper_change_log_entry, so you should remove old records first if there are lots, since deleting 2 million records with a delete statement in oracle could be bad. e.g. if you have 2560000 records in the table, this will clear out everything but 60k of them in a way that doesn't make oracle choke... note, make sure the grouper loader is not running when you do this...

```
create table change_log_entry_temp as select * from grouper_change_log_entry gcle where gcle.
SEQUENCE_NUMBER > 2500000;
truncate table grouper_change_log_entry;
insert into grouper_change_log_entry (select * from change_log_entry_temp);
commit;
analyze table grouper_change_log_entry compute statistics;
drop table change_log_entry_temp;
```

```
# number of days to retain db rows in grouper_change_log_entry. -1 is forever. default is 14
loader.retain.db.change_log_entry.days=14
```

- Add options for enabling/disabling flattened notifications. Also, remove lines that start with daily.report.syncFlatTables.

```
# Should the change log include flattened memberships?
changeLog.includeFlattenedMemberships = true

# Should the change log include flattened privileges?
changeLog.includeFlattenedPrivileges = true

# Should the change log include flattened permissions?
changeLog.includeFlattenedPermissions = true
```

- Add change log consumers for Grouper Rules and to sync Grouper instances

```
#rules consumer, needed for some of the Grouper rule types to run (e.g. flattenedMembershipRemove,
flattenedMembershipAdd)
changeLog.consumer.grouperRules.class = edu.internet2.middleware.grouper.changeLog.esb.consumer.
RuleConsumer
changeLog.consumer.grouperRules.quartzCron =

#consumer for syncing groups to other groupers
changeLog.consumer.syncGroups.class = edu.internet2.middleware.grouper.client.GroupSyncConsumer
changeLog.consumer.syncGroups.quartzCron =
```

- Rules configuration

```
#####
## Rules config
#####

# when the rules validations and daemons run. Leave blank to not run
rules.quartz.cron = 0 0 7 * * ?
```

- ESB integration configuration

```
#####
## ESB integration
#####

#changeLog.consumer.xmppTest.quartzCron =
#changeLog.consumer.xmppTest.class = edu.internet2.middleware.grouper.changeLog.esb.consumer.
EsbConsumer
#changeLog.consumer.xmppTest.elfilter = event.eventType eq 'GROUP_DELETE' || event.eventType eq
'GROUP_ADD' || event.eventType eq 'MEMBERSHIP_DELETE' || event.eventType eq 'MEMBERSHIP_ADD'
#changeLog.consumer.xmppTest.publisher.class = edu.internet2.middleware.grouper.changeLog.esb.
consumer.EsbXmppPublisher
#changeLog.consumer.xmppTest.publisher.server = jabber.school.edu
#changeLog.consumer.xmppTest.publisher.port = 5222
#changeLog.consumer.xmppTest.publisher.username = jabberuser
#changeLog.consumer.xmppTest.publisher.password = /home/whatever/pass/jabberuserEncrypted.pass
#changeLog.consumer.xmppTest.publisher.recipient = system1@school.edu
#changeLog.consumer.xmppTest.publisher.addSubjectAttributes = NETID
```

- v2.0.0: Merge ehcache.xml with ehcache.example.xml and merge grouper.ehcache.xml with grouper.ehcache.example.xml. Unless you previously customized these configuration files, you should be able to just copy the new configuration.
- v2.0.0: Add and customize grouper.client.properties based on grouper.client.example.properties.
- v2.0.0: Add email templates for Grouper rules (under conf/grouperRulesEmailTemplates).
- v2.0.0: Merge log4j.example.properties with log4j.properties to replace sync'ing of flat tables with sync'ing of point in time tables.

```
## Grouper Sync Point in Time Tables
log4j.logger.edu.internet2.middleware.grouper.misc.SyncPITTables = INFO, grouper_event

## Grouper Sync Flat Tables
#####log4j.logger.edu.internet2.middleware.grouper.misc.SyncFlatTables = INFO, grouper_event
```

- v2.0.0: Merge sources.xml with sources.example.xml
 - You can control error handling on findAll() failures

```
<!--
  You can flag a source as not throwing exception on a findAll (general search) i.e. if it is
  ok if it is down. Generally you probably won't want to do this. It defaults to true if
  omitted.

  <init-param>
    <param-name>throwErrorOnFindAllFailure</param-name>
    <param-value>false</param-value>
  </init-param>
-->
```

- Subject API changed to support [sort and search strings for subjects](#). You can use the sources.xml file to map subject attributes to sort and search fields that are stored in Grouper. Every source must be configured for at least one sort string and one search string. Most of the sources are configured in sources.xml but the internal and external sources are configured in grouper.properties. Here are the configuration updates for sources.xml.
The default for the g:gsa source:

```

<init-param>
  <param-name>subjectVirtualAttribute_0_searchAttribute0</param-name>
  <param-value>${subject.getAttributeValue('name')},${subject.getAttributeValue(
('displayName')},${subject.getAttributeValue('alternateName')}}</param-value>
</init-param>
<init-param>
  <param-name>sortAttribute0</param-name>
  <param-value>name</param-value>
</init-param>
<init-param>
  <param-name>searchAttribute0</param-name>
  <param-value>searchAttribute0</param-value>
</init-param>
<internal-attribute>searchAttribute0</internal-attribute>

```

The default for the jdbc source:

```

<init-param>
  <param-name>subjectVirtualAttribute_0_searchAttribute0</param-name>
  <param-value>${subject.name},${subjectUtils.defaultIfBlank(subject.getAttributeValue(
('LFNAME'), "")),${subjectUtils.defaultIfBlank(subject.getAttributeValue('LOGINID'),
"")},${subjectUtils.defaultIfBlank(subject.description, "")},${subjectUtils.defaultIfBlank(subject.
getAttributeValue('EMAIL'), "")}</param-value>
</init-param>
<init-param>
  <param-name>sortAttribute0</param-name>
  <param-value>LFNAME</param-value>
</init-param>
<init-param>
  <param-name>sortAttribute1</param-name>
  <param-value>LOGINID</param-value>
</init-param>
<init-param>
  <param-name>searchAttribute0</param-name>
  <param-value>searchAttribute0</param-value>
</init-param>
<internal-attribute>searchAttribute0</internal-attribute>

```

The default for the alternate jdbc source:

```

<init-param>
  <param-name>subjectAttributeColl</param-name>
  <param-value>description_lower</param-value>
</init-param>
<init-param>
  <param-name>subjectAttributeName1</param-name>
  <param-value>searchAttribute0</param-value>
</init-param>
<init-param>
  <param-name>sortAttribute0</param-name>
  <param-value>description</param-value>
</init-param>
<init-param>
  <param-name>searchAttribute0</param-name>
  <param-value>searchAttribute0</param-value>
</init-param>
<internal-attribute>searchAttribute0</internal-attribute>

```

The default for the jndi source:

```

<init-param>
  <param-name>subjectVirtualAttribute_0_searchAttribute0</param-name>
  <param-value>${subjectUtils.defaultIfBlank(subject.getAttributeValueOrCommaSeparated('uid'),
""}}, ${subjectUtils.defaultIfBlank(subject.getAttributeValueOrCommaSeparated('cn'),
""}}, ${subjectUtils.defaultIfBlank(subject.getAttributeValueOrCommaSeparated('exampleEduRegId'),
""}}</param-value>
</init-param>
<init-param>
  <param-name>sortAttribute0</param-name>
  <param-value>cn</param-value>
</init-param>
<init-param>
  <param-name>searchAttribute0</param-name>
  <param-value>searchAttribute0</param-value>
</init-param>
<internal-attribute>searchAttribute0</internal-attribute>

///Attributes you would like to display when doing a search
<attribute>cn</attribute>
<attribute>sn</attribute>
<attribute>uid</attribute>
<attribute>department</attribute>
<attribute>exampleEduRegId</attribute>

```

- For the jdbc and alternate jdbc sources, you can have exception thrown if too many subjects are returned.

```

<!-- if more than this many results are returned, then throw a too many subjects exception --
>^M
<init-param>
  <param-name>maxResults</param-name>
  <param-value>1000</param-value>
</init-param>

```

- The new configuration includes an SQL Server example.
- The new configuration includes an example of configuring an email attribute.
- v2.0.0: Get latest jars
 - Update lib/grouper/subject.jar
 - Add lib/grouper/grouperClient.jar

Grouper UI

- v2.0.2: Merge media.properties, decide if you want pager.removeFromSubjectSearch to be true or false

```

# If we should remove paging from subject search since we cant *really* page through all subjects,
# you would just be paging through the first part of the first page. IF you set this to true
# then you might want to bump up the default pagesize...
pager.removeFromSubjectSearch=false

```

- v2.0.2: Merge media.properties: this subject combobox setting should be more than a couple of the largest sources.xml page size for subjects

```

#max subjects in drop down
simpleMembershipUpdate.subjectComboboxResultSize=250

```

- v2.0.0: Merge media.properties
 - The default location for logos have changed.

```
# You may specify a logo for your organisation and for Grouper. Off-the-shelf
# your organisation logo appears on the left of the header and the Grouper logo
# appears on the right. Typically you would make the logos the same height.
image.organisation-logo=grouperExternal/public/assets/images/organisation-logo.gif
image.grouper-logo=grouperExternal/public/assets/images/grouper.gif
```

- Menu bar on the admin UI can contain a link to the Lite UI

```
menu.order=MyGroups ManageGroups CreateGroups JoinGroups AllGroups SearchSubjects SavedStems
SavedGroups SavedSubjects GroupTypes LiteUi Help
```

- If you're a wheel group member, default to 'act as admin' view?

```
# If you are a wheel group member determines if you default to 'act as admin' view
act-as-admin.default=true
```

- Security related groups:

```
#users must be in this group to invite external users to grouper
require.group.for.inviteExternalSubjects.logins=

#users must be in this group to assign/create/etc attributes in the UI (new attribute framework)
require.group.for.attributeUpdateLite.logins=
```

- Include options for [sorting and searching of subjects](#).

```
#### Member sorting and searching
# Whether to enable member sorting using sort attributes stored in Grouper.
member.sort.enabled=true

# Whether to use default sorting only and not allow users to specify which sort attribute to use.
member.sort.defaultOnly=false

# Whether to enable member searching using search attributes stored in Grouper.
member.search.enabled=true
```

- Misc settings

```
### Misc

# give more info about what is not serializable in the session
debugSessionSerialization = false
```

- Additional lite UI settings

```
grouperUi.subjectImg.sourceId.4 = grouperExternal
grouperUi.subjectImg.image.4 = user_red.png
grouperUi.subjectImg.screenEl.4 = ${subject.description}

#this source doesnt really exist, but it is the image for roles as opposed to groups
grouperUi.subjectImg.sourceId.5 = g:rsa
grouperUi.subjectImg.image.5 = group_key.png
grouperUi.subjectImg.screenEl.5 = ${grouperUiUtils.convertSubjectToLabel(subject)}
```

- Add internationalization


```
#####
## Internationalization
#####

# this should be true unless troubleshooting...
convertInputToUtf8 = true
```

◦ Various new settings due to new functionality

```
#####
## External subjects invitation
#####

# if the registration screen is enabled
externalMembers.enabledRegistration = false

#if admins should be emailed after each action, put comma separated addresses here
externalMembers.emailAdminsAddressesAfterActions =

#if you want to allow users to delete their record
externalMembers.allowSelfDelete = false

#####
## Invite external members
#####

inviteExternalMembers.groupComboboxResultSize = 200

# if the wheel group is allowed to be invited
inviteExternalMembers.allowWheelInInvite = false

# if the invitation screen is enabled
inviteExternalMembers.enableInvitation = false

#if link from admin UI
inviteExternalPeople.link-from-admin-ui = false

#if link from lite UI
inviteExternalPeople.link-from-lite-ui = false

#if admins should be emailed after each action, put comma separated addresses here
inviteExternalMembers.emailAdminsAddressesAfterActions =

#if we should allow invite by identifier
inviteExternalMembers.allowInviteByIdentifier = false

#####
## Simple permission update
#####

#max size for combobox when filtering attribute defs for permissions
simplePermissionUpdate.attributeDefComboboxResultSize = 200

#max size for combobox when filtering permission resources
simplePermissionUpdate.permissionResourceComboboxResultSize = 200

#max users in combobox when filtering
simplePermissionUpdate.subjectComboboxResultSize = 50

#number of rows to repeat headers on permissions screen
simplePermissionUpdate.repeatPermissionHeaderAfterRows = 20

#max chars in subject listing in permissions screen
simplePermissionUpdate.maxOwnerSubjectChars = 50

#####
## Simple attribute update
```

```
#####

#max size for combobox when filtering attribute defs to edit
simpleAttributeUpdate.attributeDefComboboxResultSize = 200

#repeat the header of which privilege is which every X rows
simpleAttributeUpdate.repeatPrivilegeHeaderAfterRows = 20

#max size for combobox when filtering privilege users to add
simpleAttributeUpdate.attributeDefPrivilegeUserComboboxResultSize = 200

#max size for combobox for search for members in assignment
simpleAttributeUpdate.memberComboboxResultSize = 200

#when showing assignments, this is the max number of chars before ellipses, -1 for no ellipses
simpleAttributeUpdate.maxOwnerSubjectChars = 50

#####
## Simple attribute name
#####

#max size for combobox when filtering attribute def names to edit
simpleAttributeNameUpdate.attributeDefNameComboboxResultSize = 200

#####
## Groups
#####

#max size for combobox when filtering groups to edit
simpleGroupUpdate.groupComboboxResultSize = 200

#max size for entity drop down in group privilege screen
simpleGroupUpdate.groupPrivilegeUserComboboxResultSize = 200

#####
## Directed graphs
#####

directedGraph.width = 1000
directedGraph.height = 600
```

- v2.0.0: Merge nav.properties
 - If you have enabled member sorting (member.sort.enabled) and disabled default sorting (member.sort.defaultOnly), be sure to add labels for each default sort string configured in grouper.properties (member.sort.defaultIndexOrder). Note that the labels used for the member.sort.stringX properties are based on your configuration in sources.xml and grouper.properties for member sorting and searching.

```
member.sort.string0=Name
member.sort.string1>Login Id

member.sort.change-sort-attribute=Change sort attribute

member.search.filter-members-hint=Enter search text to find members in the list:&nbsp;
member.search.filter-label=Searching for member:
member.search.search-members=Search for members
member.search.filter-clear=Clear member search
```

- There are many other updates in the nav.properties file. Mostly to define the text in all the new screens in the lite UI. Be sure to merge your copy.

Grouper WS

- v2.0.0: Merge grouper-ws.properties with grouper-ws.example.properties
 - Update the WS version for testing purposes

```
150c150
< ws.testing.version=v2_0_000
---
> ws.testing.version=v1_6_003
```

Subject API

Subject attributes are not case sensitive anymore. If implement your own source, and you do not extend SubjectImpl and BaseSourceAdaptor, then you need to make the following changes: all the Subject attribute methods are case-insensitive, you should use the SubjectCaseInsensitiveMap for attributes. The source attribute names should be toLowerCase, you should use the SubjectCaseInsensitiveSet for the Source attribute names.

Grouper Client

- v2.0.0: Merge grouper.client.properties with grouper.client.example.properties.
 - Encryption settings. See [GRP-542](#)

```
# pre grouper 2.0, the client encrypted passwords differently than the server. Now that the
client is part of the server,
# there are more reasons to be consistent. Change to false for pre-2.0 password encryption
behavior
encrypt.encryptLikeServer = true
```

- Update output templates for getPermissionAssignments and assignPermissions.

```
webService.getPermissionAssignments.output = Index: ${index}: permissionType: ${wsPermissionAssign.
permissionType}, role: ${wsPermissionAssign.roleName}, subject: ${wsPermissionAssign.sourceId} -
${wsPermissionAssign.subjectId}, attributeDefNameName: ${wsPermissionAssign.attributeDefNameName},
action: ${wsPermissionAssign.action}, allowedOverall: ${wsPermissionAssign.allowedOverall},
enabled: ${wsPermissionAssign.enabled}$newline$
webService.assignPermissions.output = Index: ${index}: permissionType: ${permissionType}, owner:
${ownerName}, permissionDefNameName: ${wsAttributeDefName.name}, action: ${wsAttributeAssign.
attributeAssignActionName}, disallowed: ${wsAttributeAssign.disallowed}, enabled:
${wsAttributeAssign.enabled}, attributeAssignId: ${wsAttributeAssign.id}, changed:
${wsAssignPermissionResult.changed}, deleted: ${wsAssignPermissionResult.deleted}$newline$
```

- Update Grouper version

```
grouperClient.output.version = 2.0.0

...

grouperClient.webService.client.version = v2_0_000
```

- Fix documentation link

```
## https://spaces.at.internet2.edu/display/Grouper/Grouper+XMPP+notifications+v1.6.0
```

- v2.0.0: Merge grouper.client.usage.txt with grouper.client.usage.example.txt.
 - Several operations have new usage

```
java -jar grouperClient.jar --operation=addMemberWs [--groupName=a:b:c] [--groupUuid=123abc] [--
subjectIds=subjId0,subjId1] [--subjectIdentifiers=subjIdent0,subjIdent1] [--subjectSources=source0,
source1] [--subjectIdsFile=fileName] [--subjectIdentifiersFile=fileName] [--
subjectSourcesFile=fileName] [--defaultSubjectSource=subjectSourceId] [--fieldName=fieldNameToAdd]
[--txType=GcTransactionType] [--includeGroupDetail=true|false] [--includeSubjectDetail=true|false]
[--subjectAttributeNames=name0,name1] [--replaceAllExisting=true|false] [--disabledTime=yyyy/mm/dd
hh:mi:ss] [--enabledTime=yyyy/mm/dd hh:mi:ss] [--addExternalSubjectIfNotFound=true|false] [--
actAsSubjectId=subjId] [--actAsSubjectIdentifier=subjIdent] [--actAsSubjectSource=source] [--
saveResultsToFile=fileName] [--outputTemplate=somePattern] [--paramName0=name0] [--
paramValue0=value1] [--paramNameX=xthParamName] [--paramValueX=xthParamValue] [--debug=true] [--
```

```
clientVersion=someVersion]
```

```
java -jar grouperClient.jar --operation=getMembersWs [--groupNames=a:b:c,a:b:d] [--groupUids=1234,abcd] [--fieldName=fieldNameToAdd] [--memberFilter=All|Immediate|NonImmediate|Effective|Composite] [--sourceIds=sourceId1,sourceId2] [--includeGroupDetail=true|false] [--includeSubjectDetail=true|false] [--subjectAttributeNames=name0,name1] [--actAsSubjectId=subjId] [--actAsSubjectIdentifier=subjIdent] [--actAsSubjectSource=source] [--saveResultsToFile=fileName] [--outputTemplate=somePattern] [--paramName0=name0] [--paramValue0=value1] [--paramNameX=xthParamName] [--paramValueX=xthParamValue] [--debug=true] [--clientVersion=someVersion] [--pointInTimeFrom=yyyy/mm/dd hh:mi:ss] [--pointInTimeTo=yyyy/mm/dd hh:mi:ss]
```

```
java -jar grouperClient.jar --operation=hasMemberWs [--groupName=a:b:c] [groupUuid=123abc] [--subjectIds=subjId0,subjId1] [--subjectIdentifiers=subjIdent0,subjIdent1] [--subjectSources=source0,source1] [--subjectIdsFile=fileName] [--subjectIdentifiersFile=fileName] [--subjectSourcesFile=fileName] [--defaultSubjectSource=subjectSourceId] [--fieldName=fieldNameToAdd] [--memberFilter=GcMemberFilter] [--includeGroupDetail=true|false] [--includeSubjectDetail=true|false] [--subjectAttributeNames=name0,name1] [--actAsSubjectId=subjId] [--actAsSubjectIdentifier=subjIdent] [--actAsSubjectSource=source] [--saveResultsToFile=fileName] [--outputTemplate=somePattern] [--paramName0=name0] [--paramValue0=value1] [--paramNameX=xthParamName] [--paramValueX=xthParamValue] [--debug=true] [--clientVersion=someVersion] [--pointInTimeFrom=yyyy/mm/dd hh:mi:ss] [--pointInTimeTo=yyyy/mm/dd hh:mi:ss]
```

```
java -jar grouperClient.jar --operation=getGroupsWs [--subjectIds=subjId0,subjId1] [--subjectIdentifiers=subjIdent0,subjIdent1] [--subjectSources=source0,source1] [--subjectIdsFile=fileName] [--subjectIdentifiersFile=fileName] [--subjectSourcesFile=fileName] [--defaultSubjectSource=subjectSourceId] [--memberFilter=GcMemberFilter] [--includeGroupDetail=true|false] [--includeSubjectDetail=true|false] [--subjectAttributeNames=name0,name1] [--actAsSubjectId=subjId] [--actAsSubjectIdentifier=subjIdent] [--actAsSubjectSource=source] [--saveResultsToFile=fileName] [--outputTemplate=somePattern] [--paramName0=name0] [--paramValue0=value1] [--paramNameX=xthParamName] [--paramValueX=xthParamValue] [--debug=true] [--clientVersion=someVersion] [--scope=some:folder:] [--stemName=stemNameToSearchIn] [--stemUuid=stemUuidToSearchIn] [--stemScope=ONE_LEVEL|ALL_IN_SUBTREE] [--enabled=A|T|F] [--pageSize=100] [--pageNumber=1] [--sortString=displayName] [--ascending=true|false] [--fieldName=members] [--pointInTimeFrom=yyyy/mm/dd hh:mi:ss] [--pointInTimeTo=yyyy/mm/dd hh:mi:ss]
```

```
java -jar grouperClient.jar --operation=groupSaveWs --name=a:b:c [--includeGroupDetail=true] [--txType=transactionType] [--saveMode=SaveMode] [--groupLookupName=a:b:c] [--groupLookupUuid=sd87f-dsf87-sdf89-df78f] [--description=theDescription] [--displayExtension=theDisplayExtension] [--createParentStemsIfNotExist=true|false] [--attributeName0=someName] [--attributeValue0=someValue] [--attributeNameX=xthName] [--attributeValueX=xthValue] [--compositeType=COMPLEMENT|INTERSECTION|UNION] [--leftGroupName=compositeLeft] [--rightGroupName=compositeRight] [--groupDetailParamName0=paramName] [--groupDetailParamValue0=paramValue] [--groupDetailParamNameX=xthName] [--groupDetailParamValueX=xthValue] [--typeNames=namesOfGroupTypes] [--actAsSubjectId=subjId] [--actAsSubjectIdentifier=subjIdent] [--actAsSubjectSource=source] [--saveResultsToFile=fileName] [--outputTemplate=somePattern] [--paramName0=name0] [--paramValue0=value1] [--paramNameX=xthParamName] [--paramValueX=xthParamValue] [--debug=true] [--clientVersion=someVersion]
```

```
java -jar grouperClient.jar --operation=stemSaveWs --name=groupName [--txType=transactionType] [--saveMode=SaveMode] [--stemLookupName=theName] [--stemLookupUuid=theUuid] [--description=theDescription] [--displayExtension=theDisplayExtension] [--createParentStemsIfNotExist=true|false] [--actAsSubjectId=subjId] [--actAsSubjectIdentifier=subjIdent] [--actAsSubjectSource=source] [--saveResultsToFile=fileName] [--outputTemplate=somePattern] [--paramName0=name0] [--paramValue0=value1] [--paramNameX=xthParamName] [--paramValueX=xthParamValue] [--debug=true] [--clientVersion=someVersion]
```

```
java -jar grouperClient.jar --operation=getAttributeAssignmentsWs --attributeAssignType=group|member|stem|any_mem|imm_mem|attr_def [--includeAssignmentsOnAssignments=true|false] [--attributeDefNames=a:b,b:c] [--attributeDefUids=1a,2b] [--attributeDefNameNames=a:b,b:c] [--attributeDefNameUids=1a,2b] [--ownerAttributeDefNames=a:b,b:c] [--ownerAttributeDefUids=1a,2b] [--ownerGroupNames=a:b:c,a:b:d] [--ownerGroupUids=1234,abcd] [--owner0SubjectId=subjId0] [--owner0SubjectIdentifier=subjIdent0] [--owner0SubjectSource=source0] [--ownerMembershipUids=abc,bcd] [--ownerStemNames=a:b,b:c] [--ownerStemUids=1a,2b] [--ownerMembershipAny0SubjectId=12] [--ownerMembershipAny0SourceId=xyz] [--ownerMembershipAny0GroupName=3c] [--ownerMembershipAny0GroupUuid=1a] [--attributeAssignUids=a:b,b:c] [--enabled=A|T|F] [--actions=read,write] [--includeGroupDetail=true|false] [--includeSubjectDetail=true|false] [--subjectAttributeNames=name0,name1] [--actAsSubjectId=subjId]
```

```

[--actAsSubjectIdentifier=subjIdent] [--actAsSubjectSource=source] [--saveResultsToFile=fileName]
[--outputTemplate=somePattern] [--paramName0=name0] [--paramValue0=value1] [--
paramNameX=xthParamName] [--paramValueX=xthParamValue] [--debug=true] [--clientVersion=someVersion]

java -jar grouperClient.jar --operation=getPermissionAssignmentsWs [--
includeAttributeAssignments=true|false] [--includeAssignmentsOnAssignments=true|false] [--
includeAttributeDefNames=true|false] [--includePermissionAssignDetail=true|false] [--
attributeDefNames=a:b,b:c] [--attributeDefUids=1a,2b] [--attributeDefNameNames=a:b,b:c] [--
attributeDefNameUids=1a,2b] [--roleNames=a:b:c,a:b:d] [--roleUids=1234,abcd] [--
subject0SubjectId=subjId0] [--subject0SubjectIdentifier=subjIdent0] [--
subject0SubjectSource=source0] [--enabled=A|T|F] [--actions=read,write] [--
includeGroupDetail=true|false] [--includeSubjectDetail=true|false] [--subjectAttributeNames=name0,
name1] [--actAsSubjectId=subjId] [--actAsSubjectIdentifier=subjIdent] [--
actAsSubjectSource=source] [--pointInTimeFrom=yyyy/mm/dd hh:mi:ss] [--pointInTimeTo=yyyy/mm/dd hh:
mi:ss] [--immediateOnly=T|F] [--permissionType=role_subject|role] [--
permissionProcessor=FILTER_REDUNDANT_PERMISSIONS|FILTER_REDUNDANT_PERMISSIONS_AND_PROCESS_LIMITS|FI
LTER_REDUNDANT_PERMISSIONS_AND_ROLES|FILTER_REDUNDANT_PERMISSIONS_AND_ROLES_AND_PROCESS_LIMITS|PROC
ESS_LIMITS] [--limitEnvVarName0=name0] [--limitEnvVarValue0=value0] [--
limitEnvVarType0=integer|decimal|date|timestamp|text|boolean|null|emptyString] [--
limitEnvVarNameX=xthName] [--limitEnvVarValueX=xthValue] [--limitEnvVarTypeX=xthType] [--
includeLimits=T|F] [--saveResultsToFile=fileName] [--outputTemplate=somePattern] [--
paramName0=name0] [--paramValue0=value1] [--paramNameX=xthParamName] [--paramValueX=xthParamValue]
[--debug=true] [--clientVersion=someVersion]

e.g.: java -jar grouperClient.jar --operation=getPermissionAssignmentsWs --
permissionType=role_subject --attributeDefNames=test:testAttributeAssignDefNameDef
output line: Index: 0: permissionType: role_subject, role: test:someRole, subject: 123456,
attributeDefNameName: test:testPermission, action: assign, allowedOverall: T, enabled: T

java -jar grouperClient.jar --operation=assignPermissionsWs --permissionType=role|role_subject --
permissionAssignOperation=assign_permission|remove_permission|replace_permissions [--
permissionDefNameNames=a:b,b:c] [--permissionDefNameUids=1a,2b] [--roleNames=a:b:c,a:b:d] [--
roleUids=1234,abcd] [--subjectRole0SubjectId=12] [--subjectRole0SubjectIdentifier=ab] [--
subjectRole0SourceId=xyz] [--subjectRole0RoleName=3c] [--subjectRole0RoleUid=1a] [--
attributeAssignUids=a:b,b:c] [--actions=read,write] [--disallowed=true|false] [--
assignmentDisabledTime=2010/03/05_17:05:13.123] [--assignmentEnabledTime=2010/03/05_17:05:13.123]
[--assignmentNotes=someNotes] [--delegatable=TRUE|FALSE|GRANT] [--includeGroupDetail=true|false]
[--includeSubjectDetail=true|false] [--subjectAttributeNames=name0,name1] [--
actAsSubjectId=subjId] [--actAsSubjectIdentifier=subjIdent] [--actAsSubjectSource=source] [--
saveResultsToFile=fileName] [--outputTemplate=somePattern] [--attributeDefNamesToReplace=a:b,b:c]
[--attributeDefUidsToReplace=1a,2b] [--actionsToReplace=read,write] [--paramName0=name0] [--
paramValue0=value1] [--paramNameX=xthParamName] [--paramValueX=xthParamValue] [--debug=true] [--
clientVersion=someVersion]

output line: Index: 0: permissionType: role, owner: a:b:c, permissionDefNameName: test:
testAttributeAssignDefName, action: assign, disallowed: T, enabled: T, attributeAssignId:
a9c83eeb78c04ae5befcea36272d318c, changed: T, deleted: F

```