Grouper Book - Management using GSH

See also Grouper Custom Template via GSH

- GrouperShell (gsh)
  - GSH operations
  - Hints and tricks
  - API Compability
  - Installation
  - Usage
  - Supported Commands
    - Grouper API methods
    - GrouperSessions
    - Group insert / update / delete
    - Groups
    - Group Types
    - New group types on folder
    - Set attribute on group
    - Member change subject
    - Memberships
    - Privileges
    - Registry
    - Stems
    - Subjects
    - System
    - Find bad memberships
    - Grouper export to GSH script
    - XML legacy
    - XML export legacy
    - XML import legacy
    - Transactions
    - Daemon
    - Loader
    - Jobs not firing in daemon
    - GrouperShell Variables (BeanShell only)
    - Membership scripts
    - Configuration in the database
    - Misc
    - External systems
    - Create a script from SQL
  - Attribute framework
    - disableLoaders.gsh (disableLoaders.gsh)
  - Rules
  - Grouper Builtin Messaging
  - Grouper messaging
  - Expression language testing
  - Example of attribute value assign to group and SQL query
  - Example of finding groups with a certain attribute value
  - Example of finding groups with a certain attribute value on metadata assignments
  - Example of finding provisioning targets for PSPNG
- Include a common GSH file
- Stem move
- Check health of database connection or run a query
- Set password using Grouper built-in authentication
- Remove all group / folder privs for a user. Remove user from groups which have a privilege on another object
- Delete all members from a subject source

This topic is discussed in the "Grouper API - Part 2" training video.

GrouperShell (gsh)

gsh is a command line shell for administering and interacting with the Grouper API. See architectural diagram. It can be used in both a batch and interactive manner. For Grouper 2.3.0 patch 72+, it is built on GroovyShell. For older versions of Grouper, it is built on Java BeanShell. The legacy BeanShell version is now deprecated, but you can switch back to it by using one of the options:

- Setting gsh.useLegacy = true in grouper.properties.
- Using a command line argument (gsh.sh -forceLegacyGsh)

GrouperShell is for Grouper admins. End users can script with the grouper client command line utility

GSH operations

NOTE: Some classes were added a later 2.5.x releases. Not all are documented as to when they were initially added.

Category	Subtype	Action	Class
Attestation	Folders	insert / update / delete	AttestationStemSave
	Groups	insert / update / delete	AttestationGroupSave
Attribute assignment	Attribute assignment	insert / update / delete	AttributeAssignToAssignmentSave (2.5.48+)
	Folders	insert / update / delete	AttributeAssignToStemSave
	Group	insert / update / delete	AttributeAssignToGroupSave
Attribute definition		insert / update / delete	AttributeDefSave
Attribute name		insert / update / delete	AttributeDefNameSave
Attribute value		insert / update / delete	AttributeAssignValueSave
Composite		insert /update / delete	CompositeSave
		finder	CompositeFinder
Email SMTP			GrouperEmail
Gc db access		gc db access	GcDbAccess
Grouper session			GrouperSession
Group		insert / update / delete	GroupSave
		finder	GroupFinder
		copy	GroupCopy
gsh		gsh template exec	GshTemplateExec
Http			GrouperHttpClient
Ldap		ldap session utils	LdapSessionUtils
Member		finder	MemberFinder
Membership		insert / update / delete	MembershipSave
		finder	MembershipFinder
Password		insert / update / delete	GrouperPasswordSave
Privilege inheritance	Attribute definitions	insert / update / delete	PrivilegeAttributeDefInheritanceSave
	Folders	insert / update / delete	PrivilegeStemInheritanceSave
	Groups	insert / update / delete	PrivilegeGroupInheritanceSave
Provisionable	Folders	finder	ProvisionableStemFinder
		insert / update / delete	ProvisionableStemSave
	Groups	finder	ProvisionableGroupFinder
		insert / update / delete	ProvisionableGroupSave
Stem		insert/update/delete	StemSave
		finder	StemFinder
		copy	StemCopy
Subject		finder	SubjectFinder
Sync data to SQL table			GcTableSyncFromData
Types	Folders	finder	GdgTypeStemFinder
		insert / update / delete	GdgTypeStemSave
	Groups	finder	GdgTypeGroupFinder
		insert / update / delete	GdgTypeGroupSave

Hints and tricks

Escape things in groovysh with single backslash. e.g.

```
attributeValueDelegate.assignValue(RuleUtils.ruleIfConditionElName(), "\${subject.sourceId != 'g:gsa'}");
```

Check for null like this

```
${someVar ?: 'valueIfNull'}
```

GSH does not like array constructors

```
FROM
Object[] row = new Object[] {emailFromOutsystems, pennkeyPerhaps};

TO
Object[] row = new Object[2];
row[0] = emailFromOutsystems;
row[1] = pennkeyPerhaps;
```

Reset the shell after an error:

:c

Escape dollars, e.g. "${something}"

```
'$' + "{something}"
```

Externalized text

```
if you add to externalized text (config in ui), just make a key, e.g. mySchoolEmailKey, then refer to it like
this
import edu.internet2.middleware.grouper.cfg.text.GrouperTextContainer;
String template = GrouperTextContainer.textOrNull("mySchoolEmailKey");
```

API Compability

gsh is now a core part of the Grouper API and so is always compatible with the current release.

Installation

When using the Grouper API source distribution, grouper.jar needs to be built before using gsh.sh for the first time:

```
cd $GROUPER_HOME
ant dist
```

Usage

For Windows use $GROUPER_HOME\bin\gsh.bat

Run gsh as an interactive shell:

```
$GROUPER_HOME/bin/gsh.sh
```

Read gsh commands from a script file:

```
$GROUPER_HOME/bin/gsh.sh /path/to/your/script.gsh
```

Run Grouper utilities:

```
$GROUPER_HOME/bin/gsh.sh <option> args: -h,              Prints this message
args: <filename>,       Execute commands in specified file
no args:                Enters an interactive shell
args: -lightWeightProfile
       Use alternate init script (classes/groovysh_lightWeight.profile)
       which has less imports and may improve startup performance
args: -nocheck,         Skips startup check and enters an
                        interactive shell
args: -runarg <command> Run command (use \\n to separate commands)
args: -main <class> [args...]
   class,               Full class name (must have main method)
   args,                args as required by main method of class
args: -initEnv [<configDir>]
       On Windows sets GROUPER_HOME and adds GROUPER_HOME/bin to path
       For *nix 'source gsh.sh' for the same result
       configDir optionally adds an alternative conf directory than
       GROUPER_HOME/conf to the classpath
args: (-xmlimport | -xmlexport | -loader | -test | -registry |
       -findbadmemberships | -ldappc | pspngAttributesToProvisioningAttributes)                          Enter
option to get additional usage for that
                        option
  -xmlimport,           Invokes XmlImporter*
                        *XML format has changed in v1.6. To import
                        the original XML format use -xmlimportold
  -xmlexport,           Invokes XmlExporter
  -loader,              Invokes GrouperLoader
  -registry,            Manipulate the Grouper schema and install
                        bootstrap data
  -test,                Run JUnit tests
  -pspngAttributesToProvisioningAttributes Copies pspng attributes to provisioning
  -findbadmemberships,  Check for membership data inconsistencies
  -ldappc,              Run the grouper ldap provisioning connector to send data to ldap
```

With argument "-lightWeightProfile", gsh will start up with an alternate boot script (groovysh_lightWeight.profile instead of the default groovysh.profile). This will perform on startup only a few Java imports, edu.internet2.middleware.grouper.* and import edu.internet2.middleware.grouper.util.*, and does not set up any help aliases and functions.

Note: you can log sql statements run from gsh by setting this in log4j.properties

```
log4j.logger.org.apache.tools.ant = WARN
```

Run SQL file

```
./gsh.sh -registry -runsqlfile subjects.sql
```

In GSH for Grouper 2.4 and above, to not print the value of every line, use this:

```
:set verbosity QUIET
```

Valid values for verbosity are DEBUG, VERBOSE, INFO (default), and QUIET.

If the temporary directory used by your JVM doesn't allow execution of executables (e.g. the directory has the noexec option set), then you may run into an error starting GSH. Try setting the following environment variable before starting GSH.

```
export GSH_JVMARGS="-Dlibrary.jansi.path=/some/other/temp/path/with/exec"
```

Environment variables that affect GSH startup:

- GROUPER_HOME: if set to a valid Grouper directory, it will use this directory. Otherwise, it will determine it based on the path to gsh
- GROUPER_CONF: if set to a valid conf directory, it will use this directory. Otherwise it will determine it based on GROUPER_HOME
- MEM_START: Override the default -Xms Java parameter (initial Java heap size)
- MEM_MAX: Override the default -Xmx Java parameter (maximum Java heap size)
- CLASSPATH: Will prepend to the constructed classpath
- GSH_JVMARGS: Additional arguments to pass to Java

- GSH_CYGWIN: (since 2.4.0 api patch 3) if set and not blank, the script will convert paths and the classpath to Windows-style, for use with Windows Java under Cygwin
- GSH_QUIET: (since 2.4.0 api patch 3) if set and not blank, will not output preliminary diagnostic information before starting Java, other than errors

Command line arg in script

```
./gsh -runarg 'userToFind="user1"\n:load "/opt/grouper/scripts/myGSHScript.gsh"'
```

Supported Commands

Grouper API methods

Any Grouper API method can be directly invoked just by referencing it, inclusive of the class in which it is defined. Methods return a java object which can be stored in a variable. For example, the following gsh session determines all of the groups to which a given subject belongs:

```
gsh 0% GrouperSession.startRootSession();
gsh 0% subj = findSubject("SD00125")
subject: id='SD00125' type='person' source='kitn-person' name='Barton, Tom'
gsh 1% sess = GrouperSession.start(subj)
edu.internet2.middleware.grouper.GrouperSession: 29c40f97-9fb0-4e45-88bc-a14877a6c9b5,'SD00125','person'
gsh 2% member = MemberFinder.findBySubject(sess, subj)
member: id='SD00125' type='person' source='kitn-person' uuid='d0fa765e-1439-4701-89b1-9b08b4ce9daa'
gsh 3% member.getGroups()
group: name='etc:sysadmingroup' displayName='Grouper Administration:SysAdmin Group' uuid='6f77fb36-b466-481a-
84a7-7af609f1ad09'
```

GrouperSessions

Group insert / update / delete

Groups

Command	Description
getGroups(name)	Find all groups with a matching naming attribute value, returns a Set of groups When using Java 1.8+ and Grouper 2.3 (later patches)+ this can be handy to print the group.getName() values for all groups that are found. getGroups("Wheel").each{it -> println "${it.getName()}"}
GroupFinder.findByName(grouperSession, name)	Find one group by name
GroupFinder.findByUuid(grouperSession, name)	Find one group by uuid

Group Types

New group types on folder

```
GrouperSession grouperSession = GrouperSession.startRootSession();
Stem stem = StemFinder.findByName(grouperSession, "test:gdg:app", true);
AttributeDefName typeMarker = AttributeDefNameFinder.findByName("etc:objectTypes:grouperObjectTypeMarker",
true);
AttributeAssign attributeAssign = stem.getAttributeDelegate().hasAttribute(typeMarker) ? stem.
getAttributeDelegate().retrieveAssignments(typeMarker).iterator().next() : stem.getAttributeDelegate().
addAttribute(typeMarker).getAttributeAssign();
attributeAssign.getAttributeValueDelegate().assignValue("etc:objectTypes:grouperObjectTypeDirectAssignment",
"true");
attributeAssign.getAttributeValueDelegate().assignValue("etc:objectTypes:grouperObjectTypeName", "app");
```

Provisionable on folder with metadata

```
    GrouperSession grouperSession = GrouperSession.startRootSession();
    AttributeAssign attributeAssignMarker = null;
```

```
    attributeAssignMarker = new AttributeAssignSave(grouperSession).assignOwnerStemName("test:chris:test").
assignNameOfAttributeDefName("etc:provisioning:provisioningMarker").save();
    new AttributeAssignSave(grouperSession).assignOwnerAttributeAssign(attributeAssignMarker).
assignNameOfAttributeDefName("etc:provisioning:provisioningDirectAssign").addValue("true").save();
    new AttributeAssignSave(grouperSession).assignOwnerAttributeAssign(attributeAssignMarker).
assignNameOfAttributeDefName("etc:provisioning:provisioningDoProvision").addValue("ADTest").save();
    new AttributeAssignSave(grouperSession).assignOwnerAttributeAssign(attributeAssignMarker).
assignNameOfAttributeDefName("etc:provisioning:provisioningStemScope").addValue("sub").save();
    new AttributeAssignSave(grouperSession).assignOwnerAttributeAssign(attributeAssignMarker).
assignNameOfAttributeDefName("etc:provisioning:provisioningTarget").addValue("ADTest").save();
    new AttributeAssignSave(grouperSession).assignOwnerAttributeAssign(attributeAssignMarker).
assignNameOfAttributeDefName("etc:provisioning:provisioningMetadataJson").addValue("{\"md_trim_prefix\":\"
whatever\",\"md_entityId\":\"theEntityId\"}").save();
```

Set attribute on group

Add a new type with an attribute, add this to group and then set its value on the group, e.g.

```
grouperSession = GrouperSession.startRootSession();
g = GroupFinder.findByName(grouperSession, "admin:loader_groups");
type = typeAdd("sync_group");
type.addAttribute(grouperSession, "sync_group", false, "sync");
groupAddType("admin:loader_groups", "sync_group");
g.setAttribute("sync", "true");
```

Command	Description
groupAddType(group name, type name)	Add type to group
groupDelType(group name, type name)	Delete type from group
groupGetTypes(group name)	Get group's types
groupHasType(group name, type name)	Check whether group had type
typeAdd(type name)	Create custom group type
typeAddAttr(type name, attr name, read, write, required)	Create custom group attribute. *read* and *write* must be an `AccessPrivilege` (e.g. `AccessPrivilege.ADMIN`)
typeAddList(type name, attr name, read, write)	Create a custom list. *read* and *write* must be an `AccessPrivilege` (e.g. `AccessPrivilege.ADMIN`).
typeDel(type name)	Delete group type
typeDelField(type name, field name)	Delete custom field from group type
typeFind(type name)	Find the group
typeGetFields(type name)	Get fields associated with the group type

Member change subject

"Member change subject" will change the subject that a member refers to. You would want to do this when a person or entity changes their id, or if they were loaded wrong in the system. If the new subject does not have a member associated with it, this is a simple case, where the subject data is put in the member object. If the new subject does have a member object, then all data in all tables that referred to the old member object, will now refer to the new member object. The old member is deleted from the member table by default, though this is an option. Generally you will want it removed, unless there is a foreign key problem where you need to do as much work as possible. In GSH you can get a dry-run report of what will be done.

The operation is potentially time consuming only when two formerly separate Subjects are being merged into one, and that the time required is to replace the memberships (and audit fields e.g. modifiedBy) of the formerly separate Subject that is being retired with new ones associated with the other Subject.

grouperSession = GrouperSession.startRootSession();
oldSubject = findSubject("10021368");
member = MemberFinder.findBySubject(grouperSession, oldSubject);

newSubject = findSubject("10021366");
member.changeSubject(newSubject);

Command	Description
member.changeSubject(newSubject);	Change the subject of the member object. If the subject is the same, its a no-op. If the new subject does not have a Member object, then the existing member object simply gets new subject information. If the new subject does have a member object, then all objects in the grouper registry which uses the old member, will be updated to the new member. Then the old member object is deleted from the registry
member.changeSubject(newSubject,!Member.DELETE_OLD_MEMBER);	Change the subject, but dont delete the old member. Do this if the way which deletes the old member doesnt work due to foreign keys. This will do all the work it can, and the rest can be manual
member.changeSubjectReport(newSubject,Member.DELETE_OLD_MEMBER);	Dont do any of the work, just print a report to the screen of what will be done. Dry-run.

Memberships

Command	Description
addComposite(group name, composite type, left group name, right group name)	Add composite membership. e.g. CompositeType.UNION
addMember(group name, subject id)	Add member to the members list for the group.
addMember(group name, subject id, field)	Add member to the specified list for the group.
delComposite(group name)	Delete composite membership from group
delMember(group name, subject id)	Delete member from the members list for the group
delMember(group name, subject id, field)	Delete member from the specified list for the group
getMembers(group name)	Get members of group
hasMember(group name, subject id)	Check whether subject is member of the members list
hasMember(group name, subject id, field)	Check whether subject is member of the specified list
GrouperSession grouperSession = GrouperSession.startRootSession(); Group group = GroupFinder.findByName(grouperSession, "a:b:c", true); group.addMember(SubjectFinder.findByIdAndSource("someId", "sourceId", true), false);	Add member with subjectId and sourceId
GrouperSession grouperSession = GrouperSession.startRootSession(); Group group = GroupFinder.findByName(grouperSession, "a:b:c", true); group.addMember(SubjectFinder.findByIdentifierAndSource("someIdentifier", "sourceId", true), false);	Add member with subjectIdentifier and sourceId

Privileges

Command	Description
grantPriv(group name, subject id, privilege)	Grant privilege on group. *privilege* must be an *AccessPrivilege* (e.g. `AccessPrivilege.ADMIN`)
grantPriv(stem name, subject id, privilege)	Grant privilege on stem. *privilege* must be a *NamingPrivilege* (e.g. `NamingPrivilege.STEM`)
hasPriv(group name, subject id, privilege)	Check whether subject has privilege on group. *privilege* must be an *AccessPrivilege* (e.g. `AccessPrivilege.ADMIN`)
hasPriv(stem name, subject id, privilege)	Check whether subject has privilege on strem. *privilege* must be a *NamingPrivilege* (e.g. `NamingPrivilege.STEM`)
revokePriv(group name, subject id, privilege)	Revoke privilege on group. *privilege* must be an *AccessPrivilege* (e.g. `AccessPrivilege.ADMIN`)
	Revoke privilege on stem. *privilege* must be a *NamingPrivilege* (e.g. `NamingPrivilege.STEM`)

revokePriv(stem name, subject id, privilege)	

Registry

Command	Description
registryInitializeSchema()	Will generate schema DDL for the DB, and wont drop before creating, will not run script
registryInitializeSchema(registryInitializeSchema.DROP_THEN_CREATE)	generate DDL for the DB, dropping existing tables, will not run script
registryInitializeSchema.WRITE_AND_RUN_SCRIPT)	generate DDL for the DB, not dropping, but will run the script after writing it to file
registryInitializeSchema(registryInitializeSchema.DROP_THEN_CREATE | registryInitializeSchema.WRITE_AND_RUN_SCRIPT)	generate DDL for the DB, drop existing grouper tables, and run the script after writing it to file
resetRegistry()	Restore registry to default state(delete data from all tables, install defaults)
registryInstall()	If the default Grouper data is not there, it will be added (e.g. root stem, default fields, etc)

Stems

Command	Description
addRootStem (extension, displayExtension)	Add top-level stem to the registry
addStem(parent stem name, extension, displayExtension)	Add stem to registry
delStem(stem name)	Delete stem from registry
obliterateStem(stem name, testOnlyBoolean, deleteFromPointInTime Boolean) (Grouper v2. 0.2+)	Delete stem, and subobjects. If testonly (true|false), then only print a report. This is not supported when deleteFromPointInTime is true. If deleteFromPointInTime (true|false), then delete from point in time as well. Otherwise, point in time records are not deleted. Note that point in time data can only be deleted after the actual objects have been deleted and those deletions have been processed by the changeLogTempToChangeLog job, which runs once a minute by default with the Grouper Daemon. So when you call obliterateStem(name, false, true), it will first obliterate the actual stem, then sleep and keep checking if the changeLogTempToChangeLog job has completed. When it completes, it will obliterate from the point in time data. GrouperSession must be open before calling...
getStemAttr(stem name, attr)	Get value of stem attribute
getStems(name)	Find all stems with a matching naming attribute value, returns a Set of stems
setStemAttr(stem name, attr, value)	Set value of stem attribute
StemFinder. findByName (grouperSession, name)	Find one stem by name
StemFinder.findByUuid (grouperSession, uuid)	Find one stem by uuid

Delete stem and subcontents	`grouperSession = GrouperSession.startRootSession();`
	`stem = StemFinder.findByName(grouperSession, "a");`
	`for(child : stem.getChildGroups(Stem.Scope.SUB)) { System.out.println("deleting: " +`
	`child.getName()); child.delete(); }`
	`stemList = new ArrayList(stem.getChildStems(Stem.Scope.SUB));`
	`Collections.sort(stemList);`
	`Collections.reverse(stemList);`
	`for(childStem : stemList) { System.out.println("deleting: " + childStem.getName());`

```
childStem.delete(); }
stem.delete();
```

Subjects

Command	Description
addSubject(id, type, name)	Add local subject to registry. You need the jdbc source for this to work. The type parameter describes the type of subject (e.g. "person"), and is required non-null even though there are few useful api methods to query it. In 2.4.0.api.41+ patch, this will also create the id, name, description, and loginid attribute (unless grouper.properties create.attributes.when.creating.registry.subjects is false)
RegistrySubject.addOrUpdate (grouperSession, id, type, name, nameAttributeValue, loginid, description, email)	In 2.4.0.api.41+ patch, add a registry subject like addSubject, but specify the attribute values of name, loginid, etc e.g. RegistrySubject.addOrUpdate(grouperSession, "someTestSubject", "person", "Some Testsubject", "Name Some Test Subject", "stsub", "Some Testsubject - employee - also alumni", "some@test.subject");
RegistrySubject.find(id, errorOnNotFound)	In 2.4.0.api.41+ patch, get a registry subject e.g. registrySubject = RegistrySubject.find("someTestSubject", false);
registrySubject.delete (grouperSession)	In 2.4.0.api.41+ patch, delete a registry subject e.g. registrySubject.delete(grouperSession);
RegistrySubjectAttribute. addOrUpdate(subjectId, attributeName, attributeValue)	In 2.4.0.api.41+ patch, add or update a registry subject attribute
registrySubjectAttribute.delete()	In 2.4.0.api.41+ patch, delete an attribute value
findSubject(idOrIdentifier)	Find a subject by id or identifier
findSubject(idOrIdentifier, type)	Find a subject by id or identifier; type is a deprecated parameter that is ignored
findSubject(idOrIdentifier, type, source)	Find a subject by id or identifier for a specific subject source; type is a deprecated parameter that is ignored
getSources()	Find all Subject sources
grouperSession = GrouperSession. startRootSession(); SubjectFinder.findAll(searchString, source);	Find all subjects in a source by search string
grouperSession = GrouperSession. startRootSession(); SubjectFinder.findByIdAndSource (id, source, exceptionIfNull); SubjectFinder.findByIdAndSource ("12345", "jdbc", true);	Find a subject by id in a certain source
grouperSession = GrouperSession. startRootSession(); SubjectFinder. findByIdentifierAndSource (identifier, source, exceptionIfNull); SubjectFinder.findByIdentifierAndSource("jsmith", "jdbc", true);	Find a subject by identifier in a certain source
grouperSession = GrouperSession. startRootSession(); SubjectFinder. findByIdOrIdentifierAndSource (idOrIdentifier, source, exceptionIfNull); SubjectFinder. findByIdOrIdentifierAndSource ("jsmith", "jdbc", true);	Find a subject by id or identifier in a certain source
add test subjects to registry (e.g. test.subject.0 through 9)	grouperSession = GrouperSession.startRootSession(); new RegistryReset()._addSubjects();
Edit subject (in this case name)	RegistrySubject registrySubject = GrouperDAOFactory.getFactory().getRegistrySubject().find("user1a", "person", true); registrySubject.setName("New name"); HibernateSession.byObjectStatic().update(registrySubject);

add a subject application principal with attributes (GSH)	`String principal = "someApp";`
	`String email = null;`
	
	`GrouperSession grouperSession = GrouperSession.startRootSession();`
	
	`addSubject(principal, "application", principal);`
	`HibernateSession.bySqlStatic().executeSql("insert into subjectattribute (subjectId, name, value, searchValue) values (?, ?, ?, ?)", GrouperUtil.toListObject(new Object[]{principal, "description", principal, principal.toLowerCase()}));`
	`if (email != null){ HibernateSession.bySqlStatic().executeSql("insert into subjectattribute (subjectId, name, value, searchValue) values (?, ?, ?, ?)", GrouperUtil.toListObject(new Object[]{principal, "email", email, email.toLowerCase()}));}`
	`HibernateSession.bySqlStatic().executeSql("insert into subjectattribute (subjectId, name, value, searchValue) values (?, ?, ?, ?)", GrouperUtil.toListObject(new Object[]{principal, "loginid", principal, principal}));`
	`HibernateSession.bySqlStatic().executeSql("insert into subjectattribute (subjectId, name, value, searchValue) values (?, ?, ?, ?)", GrouperUtil.toListObject(new Object[]{principal, "name", principal, principal}));`

remove a subject with attributes (GSH)	`String principal = "someApp";`
	`String email = null;`
	
	`GrouperSession grouperSession = GrouperSession.startRootSession();`
	
	`HibernateSession.bySqlStatic().executeSql("delete from subjectattribute where subjectId = ?", GrouperUtil.toListObject(new Object[]{principal}));`
	`HibernateSession.bySqlStatic().executeSql("delete from subject where subjectId = ?", GrouperUtil.toListObject(new Object[]{principal}));`

subject api diagnostics	grouper.properties (temporarily)
	
	`gsh.exitOnSubjectCheckConfigProblem = false`
	
	gsh
	
	`GrouperSession.startRootSession();`
	`new edu.internet2.middleware.grouper.grouperUi.serviceLogic.SubjectSourceDiagnostics().assignSourceId("SMUPerson_DEV").assignSubjectId("empl1").assignSubjectIdentifier("netid@school.edu").assignSearchString("em").subjectSourceDiagnosticsFromGsh()`
	`===>`
	`SUCCESS: Found subject by id in 37ms: 'empl1'`
	`with SubjectFinder.findByIdAndSource("empl1", "SMUPerson_DEV", false)`
	`SUCCESS: Subject id in returned subject matches the subject id searched for: 'empl1'`
	`WARNING: No subject found by identifier in 14ms: 'netid@school.edu'`
	`with SubjectFinder.findByIdentifierAndSource("netid@school.edu", "SMUPerson_DEV", false)`

System

Command	Description
sqlRun(file)	Execute each line of a sql file, just like ant would. This can run the files generated by registryInitializeSchema()
sqlRun(string)	Executes a single sql statement

:exit	Terminate shell
help()	Display usage information
p(command)	Pretty print results.
:quit	Terminate shell
version()	Return version information

Find bad memberships

This command will find membership records in the database which are invalid, and prints them on the screen, along with a GSH script that will fix the memberships.

For more information, see Bad Membership Finder Utility

Command	Description
findBadMemberships()	complete findBadMemberships run

Grouper export to GSH script

See page: Grouper export to a GSH script

XML legacy

Command	Description
xmlFromFile(filename)	Load registry from XML in file
xmlFromString(xml)	Load registry from XML in string
xmlFromURL(url)	Load registry from XML at URL
xmlToFile(filename)	Exports registry to file
xmlToString()	Exports registry to string.
xmlUpdateFromFile(filename)	Update registry from XML in file
xmlUpdateFromString(xml)	Update registry from XML in string
xmlUpdateFromURL(url)	Update registry from XML at URL

XML export legacy

There is an object: XmlExport which has various chaining methods, which should be ended with an exportTo() method. You can export to file or string.

For more information, see Import-Export

Command	Description
XmlExport xmlExport.stem(stem)	The stem to export. Defaults to the ROOT stem.
XmlExport xmlExport.group(group)	The group to export
XmlExport xmlExport.relative(boolean)	If group or stem specified do not export parent Stems.
XmlExport xmlExport.includeParent(boolean)	If group specified, export from the parent stem
XmlExport xmlExport.childrenOnly(boolean)	If stem specified, export child stems and groups only - not the specified stem
XmlExport xmlExport.userProperties(file)	Properties file for extra settings for import
XmlExport xmlExport.grouperSession(grouperSession)	Operate within a certain grouper session (defaults to root session)
void xmlExport.exportToFile(file)	Export to an XML file
void xmlExport.exportToString(string)	Export to an XML string

Examples:

```
gsh 1% new XmlExport().exportToFile(new File("c:/temp/export.xml"))
```

```
gsh 1% grouperSession = GrouperSession.start(SubjectFinder.findById("mchyzer"));
gsh 2% stem = StemFinder.findByName(grouperSession, "aStem");
gsh 3% new XmlExport().stem(stem).relative(true).userProperties(new File("C:/temp/some.props")).grouperSession
(grouperSession).exportToFile(new File("c:/temp/export.xml"));
```

-or- (without chaining)

```
gsh 3% xmlExport = new XmlExport();
gsh 4% xmlExport.stem(stem);
gsh 5% xmlExport.grouperSession(grouperSession);
gsh 6% xmlExport.exportToFile(new File("c:/temp/export.xml"))
```

XML import legacy

There is an object: XmlImport which has various chaining methods, which should be ended with an importFrom() method. You can import from file, string, or url.

For more information, see Import-Export

Command	Description
XmlImport xmlImport.stem(stem)	The Stem into which data will be imported. Defaults to the ROOT stem.
XmlImport xmlImport.updateList(boolean)	XML contains a flat list of Stems or Groups which may be updated. Missing Stems and Groups are not created.
XmlImport xmlImport.userProperties(file)	Properties file for extra settings for import
XmlImport xmlImport.grouperSession(grouperSession)	Operate within a certain grouper session (defaults to root session)
XmlImport xmlImport.ignoreInternal(boolean)	Ignore internal attributes, including group and stem uuids.
void xmlImport.importFromFile(file)	Import from an XML file
void xmlImport.importFromString(string)	Import from an XML string
void xmlImport.importFromUrl(url)	Import XML from a URL

Examples:

```
gsh 1% new XmlImport().importFromFile(new File("c:/temp/export.xml"))
```

```
gsh 1% grouperSession = GrouperSession.start(SubjectFinder.findById("mchyzer"));
gsh 2% stem = StemFinder.findByName(grouperSession, "aStem");
gsh 3% new XmlImport().stem(stem).updateList(true).userProperties(new File("C:/temp/some.props")).grouperSession
(grouperSession).importFromUrl(new URL("http://whatever.xml"));
```

-or- (without chaining)

```
gsh 3% xmlImport = new XmlImport();
gsh 4% xmlImport.stem(stem);
gsh 5% xmlImport.grouperSession(grouperSession);
gsh 6% xmlImport.importFromFile(new File("c:/temp/export.xml"))
```

Transactions

Transactions facilitate all commands succeeding or failing together, and perhaps some level of repeatable reads of the DB (depending on the DB). If there is an open transaction and an exception is thrown in a command, GSH will shut down so that subsequent commands will not execute outside of a transaction.

Command	Description

help("transaction")	print help information
transactionStatus()	print the list of nested transactions
transactionStart("<GrouperTransactionType>")	start a transaction, or make sure one is already started Can use: "READONLY_OR_USE_EXISTING", "NONE", "READONLY_NEW", "READ_WRITE_OR_USE_EXISTING", "READ_WRITE_NEW"
transactionCommit("<GrouperCommitType>")	commit a transaction Can use: "COMMIT_NOW", "COMMIT_IF_NEW_TRANSACTION
transactionRollback("<GrouperRollbackType>")	rollback a transaction Can use: "ROLLBACK_NOW", "ROLLBACK_IF_NEW_TRANSACTION
transactionEnd()	end a transaction Note if it was read/write, and not committed or rolled back, this will commit and end

Daemon

You can schedule daemon jobs (and in UI) in v2.5.23+

```
GrouperSession.startRootSession();
edu.internet2.middleware.grouper.app.loader.GrouperLoader.scheduleJobs();
```

Enable a job (OTHER_JOB_grouperLoaderJexlScriptFullSync is the job name)

```
edu.internet2.middleware.grouper.app.loader.GrouperLoader.schedulerFactory().getScheduler().resumeJob(new org.
quartz.JobKey("OTHER_JOB_grouperLoaderJexlScriptFullSync"));
```

Loader

Above, it describes how you can kick off the loader in daemon mode. You can also execute one job with:

Command	Description

grouperSession = GrouperSession.startRootSession();
loaderGroup = GroupFinder.findByName(grouperSession, "stem:group");
loaderRunOneJob(loaderGroup);
	Kick off the loader for one group (configured by group attributes)

loaderRunOneJob("MAINTENANCE_cleanLogs");	Kick off the loader by job name
loaderRunOneJob("CHANGE_LOG_changeLogTempToChangeLog");	Move change log entries from the temp table to the real table
loaderRunOneJob("CHANGE_LOG_consumer_grouperRules"); loaderRunOneJob("MAINTENANCE__rules");	Run the Grouper Rules daemon (the changelog or full version)
loaderRunOneJob("CHANGE_LOG_consumer_test");	Run a change log consumer
GrouperLoaderType.validateAndScheduleSqlLoad(group, null, false)	Schedule SQL job
GrouperLoaderType.validateAndScheduleLdapLoad(attributeAssign, null, false)	Schedule LDAP job
GrouperLoaderType.scheduleAttributeLoads();	Schedule all attribute loader jobs

This query (in Oracle) will find jobs with no success in the last day and make a gsh script:

```
select distinct 'loaderRunOneJob("' || job_name || '");' as script
from grouper_loader_log gll where started_time > sysdate-1 and status != 'SUCCESS'
and gll.job_name not like 'subjobFor%'
and not exists (select 1 from grouper_loader_log gll2 where gll2.started_time > sysdate-1
 and gll2.status = 'SUCCESS' and gll2.job_name = gll.job_name)
```

HSQLDB (similar example)

```
select distinct job_name from grouper_loader_log gll where started_time > CURRENT_DATE - 1 DAY and status !=
'SUCCESS' and gll.job_name not like 'subjobFor%'
AND NOT EXISTS (select job_name from grouper_loader_log gll2 where gll2.started_time > CURRENT_DATE - 1 DAY and
gll2.status = 'SUCCESS' and gll2.job_name = gll.job_name)
```

v1.6+ loader

Command	Description
loaderRunOneJobAttr(attirbuteDef)	Run an attribute definition loader job

You can run the loader as a linux service

Jobs not firing in daemon

1. Stop all daemons
2. Run these sqls from a file: fixLoaderScheduler.sql

```
UPDATE GROUPER_QZ_TRIGGERS SET TRIGGER_STATE = 'WAITING';
DELETE FROM GROUPER_QZ_FIRED_TRIGGERS;
commit;
```

```
gsh.sh -registry -runsqlfile fixLoaderScheduler.sql
```

3. Restart daemons

It took a couple of hours to catch up on a few days of changes, but it seems to be back to normal. Thanks again, guys!

GrouperShell Variables (BeanShell only)

gsh has several variables that can be set to modify runtime behavior

Variable	Description
GSH_DEBUG	Stack traces will be printed upon failure if true
GSH_DEVEL	Summaries of returned objects are not automatically printed if true
GSH_TIMER	Prints time spent evaluating each command if true

Example:

```
gsh 4% GSH_DEVEL = true
gsh 5% subj = findSubject("SD00125")
gsh 6% sess = GrouperSession.start(subj)
gsh 7% member = MemberFinder.findBySubject(sess, subj)
gsh 8% p(member.getGroups())
group: name='etc:sysadmingroup' displayName='Grouper Administration:SysAdmin Group' uuid='6f77fb36-b466-481a-
84a7-7af609f1ad09'
```

Membership scripts

```
# (1) Print tab-separated summary of all group members, and flags for direct, indirect, or both
# Depending on the results, you could use the data to create a scrutinized list of Ids to delete, then import
it and delete in a loop

me = SubjectFinder.findByIdentifierAndSource("my-username", "pid", true);
session = GrouperSession.start(me);
// OR: session = GrouperSession.startRootSession(True)

group = GroupFinder.findByName(session, "tmp:my:group", true);
```

```
effectiveMembers = group.getEffectiveMembers();
immediateMembers = group.getImmediateMembers();

System.out.println(String.join("\t", "id", "name", "Effective", "Immediate"));

for (Member m: group.getMembers()) {
    System.out.print(m.getSubject().getId() + "\t" + m.getSubject().getName() + "\t");
    System.out.print(effectiveMembers.contains(m).toString() + "\t");
    System.out.println(immediateMembers.contains(m).toString() + "\t");
}



# (2) Get the immediate and effective members for a specific source ("pid" in this example), intersect them to
find the redundant ones
# This has a dryRun flag, so you can test first

sources = new HashSet<Source>()
sources.add(SourceManager.getInstance().getSource("pid"))

effectiveUsers = group.getEffectiveMembers(Group.getDefaultList(), sources, null)
immediateUsers = group.getImmediateMembers(Group.getDefaultList(), sources, null)

# use retainAll() to find the intersection; i.e., users both as effective and immediate member
immediateUsers.retainAll(effectiveUsers)

System.out.println("There are " + immediateUsers.size() + " users having both direct + indirect memberships");

dryRun = true

for (Member m: immediateUsers) {
    if (dryRun) {
        System.out.println("Ok to delete " + m.getSubject().getId());
    } else {
        System.out.println("Deleting " + m.getSubject().getId());
        group.deleteMember(m, false);
    }
}

# (3) Get the groups this subject is a member of. Note that a group is a kind of subject, and has a toSubject()
method to convert it.

import edu.internet2.middleware.grouper.membership.MembershipSubjectContainer

GrouperSession grouperSession = GrouperSession.startRootSession();

Group group = GroupFinder.findByName(grouperSession, "test:testGroup", true);
Subject subject = g.toSubject();

Set<MembershipSubjectContainer> msc = new MembershipFinder().addSubject(subject).findMembershipResult().
getMembershipSubjectContainers();

for (MembershipSubjectContainer membershipSubjectContainer : msc) { println(membershipSubjectContainer.
getGroupOwner().getName());}

//Note there are a few other options for the search. Add these to the MembershipFinder method chain before
calling findMembershipResult():
//  - search immediate, effective, etc. (needs to import MembershipType)
import edu.internet2.middleware.grouper.membership.MembershipType
membershipFinder.assignMembershipType(MembershipType.IMMEDIATE) // options are
IMMEDIATE|NONIMMEDIATE|EFFECTIVE|COMPOSITE
//  - retrieve specific groups based on pattern
membershipFinder.assignScope("%:test:%")
//  - Enabled status -- true means enabled only, false, means disabled only, and null means all
membershipFinder.assignEnabled(false)
// For other methods, refer to the Javadoc at https://software.internet2.edu/grouper/doc/master/grouper/apidocs
/index.html?edu/internet2/middleware/grouper/MembershipFinder.html
```

Configuration in the database



In v2.4.0 ui patch #56+, or 2.5 versions before 2.5.51, replace edu.internet2.middleware.grouper.cfg.dbConfig.GrouperDbConfig() in the script below with edu.internet2.middleware.grouper.grouperUi.beans.config.GrouperDbConfig(). Also, GSH **must** be run from the UI in WEB-INF/bin.

In 2.5.51+, use class edu.internet2.middleware.grouper.cfg.dbConfig.GrouperDbConfig for ADD / DELETE and edu.internet2.middleware.grouper.grouperUi.beans.config.GrouperDbConfigImport for IMPORT as in the example.

```
GrouperSession.startRootSession();

ADD
new edu.internet2.middleware.grouper.cfg.dbConfig.GrouperDbConfig().configFileName("grouper.properties").
propertyName("abc").value("123").store();

DELETE
new edu.internet2.middleware.grouper.cfg.dbConfig.GrouperDbConfig().configFileName("grouper.properties").
propertyName("abc").delete();
GrouperDAOFactory.getFactory().getConfig().findAll(ConfigFileName.GROUPER_LOADER_PROPERTIES, null, "changeLog.
consumer.pspng_oneprod.groupCreationLdifTemplate").iterator().next().delete();

IMPORT
new edu.internet2.middleware.grouper.grouperUi.beans.config.GrouperDbConfigImport().configFilePath("d:/temp/temp
/grouper.properties").store();
```

Misc

Note: you can use the MorphString class to encrypt and update a password field, but it will be saved in the local GSH command history:

```
import edu.internet2.middleware.morphString.Morph
new edu.internet2.middleware.grouper.cfg.dbConfig.GrouperDbConfig().configFileName("grouper-loader.properties").
propertyName("loader.myApp.password").value(Morph.encrypt('xxxxxxxx')).store();
```

You can also encrypt the password outside of GSH:

```
java -cp "lib/*:classes" edu.internet2.middleware.morphString.Encrypt
Type the string to encrypt (note: pasting might echo it back):
The encrypted string is: ca8a15be4ad0fb45c6f1b3ca0cfd9c9e
```

v2.0: to sync up the point in time tables with regular tables. In v2.5.47+, run the one-time OTHER_JOB_syncAllPitTables daemon. In a prior version, you can run this.

```
new edu.internet2.middleware.grouper.misc.SyncPITTables().syncAllPITTables()
```

To create missing group sets:

```
new edu.internet2.middleware.grouper.misc.AddMissingGroupSets().addAllMissingGroupSets();
```

Delete memberships not in transaction

```
grouperSession = GrouperSession.startRootSession();
group = GroupFinder.findByName(grouperSession, "test:testGroup3", true);
for (membership : group.getImmediateMemberships()) {membership.delete();}
group.delete();
```

Note: in v2.4.0 patch 91+ (unreleased at the time of writing) you can use gsh to do simple sql tests through jdbc loader connection

```
gcDbAccess = new edu.internet2.middleware.grouperClient.jdbc.GcDbAccess();
// "loaderConnection" is the string used in the grouper-loader.properties ( Example: db.warehouse.url -->
"warehouse")
gcDbAccess.connectionName("loaderConnection").sql("select count(1) from test1").select(int.class);
```

```
List results = new edu.internet2.middleware.grouperClient.jdbc.GcDbAccess().sql("select name, id from
grouper_groups").selectList(Object[].class);
for (Object[] row : results) { System.out.println(row[0] + ", " + row[1]);}
```

See the WIKI for running the Grouper Report manually

External systems

Test all external systems

```
import java.util.List;

import edu.internet2.middleware.grouper.app.externalSystem.GrouperExternalSystem;
import edu.internet2.middleware.grouper.app.gsh.template.GshTemplateOutput;
import edu.internet2.middleware.grouper.app.gsh.template.GshTemplateV2;
import edu.internet2.middleware.grouper.app.gsh.template.GshTemplateV2input;
import edu.internet2.middleware.grouper.app.gsh.template.GshTemplateV2output;
import edu.internet2.middleware.grouper.util.GrouperUtil;

public class Test77externalSystemTest extends GshTemplateV2 {

  @Override
  public void gshRunLogic(GshTemplateV2input gshTemplateV2input, GshTemplateV2output gshTemplateV2output) {

    GshTemplateOutput gsh_builtin_gshTemplateOutput = gshTemplateV2output.getGsh_builtin_gshTemplateOutput();

    gsh_builtin_gshTemplateOutput.assignRedirectToGrouperOperation("NONE");

    List<GrouperExternalSystem> grouperExternalSystems = GrouperExternalSystem.
retrieveAllGrouperExternalSystems();

    for (GrouperExternalSystem grouperExternalSystem : grouperExternalSystems) {
      try {
        List<String> errors = grouperExternalSystem.test();
        if (GrouperUtil.length(errors) == 0) {
          gsh_builtin_gshTemplateOutput.addOutputLine("Success: extenal system '" + grouperExternalSystem.
getConfigId() + "' passed its test");
          continue;
        }
        for (String error : errors) {
          gsh_builtin_gshTemplateOutput.addOutputLine("error", "Error: extenal system '" +
grouperExternalSystem.getConfigId() + "': " + error);
        }
      } catch (Exception e) {
        gsh_builtin_gshTemplateOutput.addOutputLine("error", "Error: extenal system '" + grouperExternalSystem.
getConfigId() + "' failed: " + e.getMessage());
      }
    }

  }

}
```

Create a script from SQL

Here is an example to remove access from someone... run a SQL to generate a GSH script, e.g. in oracle:

```
set linesize 1000;
set pagesize 1000;
select 'delMember("' || gmlv.GROUP_NAME || '", "' || gmlv.SUBJECT_ID || '");' as script
from grouper_memberships_lw_v gmlv where subject_id = '12345678' and gmlv.LIST_NAME = 'members';
```

Put that script in a text editor and remove extra whitespace (probably optional), and add this to the beginning:

```
grouperSession = GrouperSession.startRootSession();
```

Look at it and remove lines that dont apply... then run in GSH

```
[appadmin@lorenzo bin]$ ./gsh.sh remove.script
```

Here is a more complicated example. I want all groups in a certain folder which do not have an ADMIN privilege assigned to my application service principal, to assign that privilege. Here is the query for oracle:

```
select 'grantPriv("' || gg.name || '", "someid/server.school.edu", AccessPrivilege.ADMIN);' as script
from grouper_groups gg where gg.name like 'school:apps:appName:spaces:%'
and not exists
(select (1) from grouper_memberships_lw_v gmlv where gg.name = gmlv.group_name and list_name = 'admins'
and gmlv.subject_id = 'someid/server.school.edu');
```

Here is an example of deleting memberships for a user in oracle, dont forget at top of script to add grouperSession = GrouperSession.startRootSession():

```
set linesize 1000;
set pagesize 1000;
select 'delMember("' || gg.name || '", "' || gm.subject_id || '");'
  as script
from grouper_memberships_all_v gmav, grouper_fields gf, grouper_groups gg, grouper_members gm
where GMAV.FIELD_ID = GF.ID and gm.subject_id = '12345678' and GF.name = 'members'
and GMAV.OWNER_GROUP_ID = gg.ID and GMAV.MEMBER_ID = GM.ID and GMAV.DEPTH = 0
```

Here is an example of removing privileges from a user on groups in oracle, dont forget at top of script to add grouperSession = GrouperSession. startRootSession():

```
set linesize 1000;
set pagesize 1000;
select 'revokePriv("' || gmlv.group_name || '", "' || gmlv.subject_id || '", AccessPrivilege.' ||
case
when gmlv.LIST_NAME = 'admins' then 'ADMIN'
when gmlv.LIST_NAME = 'readers' then 'READ'
when gmlv.LIST_NAME = 'viewers' then 'VIEW'
when gmlv.LIST_NAME = 'updaters' then 'UPDATE'
when gmlv.LIST_NAME = 'optins' then 'OPTIN'
when gmlv.LIST_NAME = 'optouts' then 'OPTOUT'
else gmlv.LIST_NAME
end  || ');'
  as script
from grouper_memberships_lw_v gmlv where subject_id = '12345678' and GMLV.LIST_TYPE = 'access'
```

This oracle script will remove privileges on folders for a certain user, dont forget at top of script to add grouperSession = GrouperSession. startRootSession():

```
set linesize 1000;
set pagesize 1000;
select 'revokePriv("' || gs.name || '", "' || gm.subject_id || '", NamingPrivilege.' ||
case
when gf.NAME = 'stemmers' then 'STEM'
when gf.NAME = 'creators' then 'CREATE'
else gf.NAME
end  || ');'
  as script
```

```
from grouper_memberships_all_v gmav, grouper_fields gf, grouper_stems gs, grouper_members gm
where GMAV.FIELD_ID = GF.ID and gm.subject_id = '12345678' and GF.type = 'naming'
and GMAV.OWNER_STEM_ID = GS.ID and GMAV.MEMBER_ID = GM.ID
```

Example of copying memberships and enabled/disabled dates from one group to another (postgres/oracle)

```
select 'new MembershipSave().assignGroupName("test:testGroup2").assignMemberId("' || gm.member_id || '")'
  || (case when gm.enabled_timestamp is not null then '.assignImmediateMshipEnabledTime(' || cast(gm.
enabled_timestamp as varchar) || 'L)' else '' end)
  || (case when gm.disabled_timestamp is not null then '.assignImmediateMshipDisabledTime(' || cast(gm.
disabled_timestamp as varchar) || 'L)' else '' end)
  || '.save();' as script
  from grouper_memberships gm, grouper_groups gg, grouper_fields gf
where gm.field_id = gf.id and gm.owner_group_id = gg.id
and gf.name = 'members' and gg.name = 'test:testGroup1'
```

Attribute framework

Create a permission and configure action list:

```
grouperSession = GrouperSession.startRootSession();
attributeDef = new AttributeDefSave(grouperSession).assignName("stem2:sub:c").assignToEffMembership(true).
assignToGroup(true).assignAttributeDefType(AttributeDefType.perm).assignCreateParentStemsIfNotExist(true).
save();
attributeDef.getAttributeDefActionDelegate().configureActionList("read,write");
```

Retrieve assignments for the attribute "school:attr:students:artsAndSciences"

```
attributeDefName = AttributeDefNameFinder.findByName("school:attr:students:artsAndSciences", true);
group.getAttributeDelegate().retrieveAssignments(attributeDefName);
```

disableLoaders.gsh (disableLoaders.gsh)

The following script will print to standard output (not saved as files) two scripts.

- One to disable all loader jobs (AKA: "DISABLE ALL SCHEDULES").
- A second one to re-enabled them (AKA: "RESTORE OLD SCHEDULES") .

Note: The disableLoaders.gsh script does **not change the state** of the loader jobs. Rather it **only** prints (outputs) GSH scripts that you can later execute to do disable/enable for the jobs on the system at the time.

Note: After running either of the scripts that are output, you **need to restart** all grouper **daemon instances** to make the changes effective.(So you *might* choose to stop them before running the "DISABLE" or "RESTORE" script.That order is not strictly required.)

Note well: The method used to "disable" the jobs is to alter the quartz schedule for the job to be a fixed time in the distant future. (specifically: "0 0 0 1 1 ? 3000") So the "RESTORE OLD SCHEDULES" script is the **only record** of what the **orginal scheduled** values were. **Don't lose it.**

Note this works in the new GSH. To use in legacy GSH, take the set verbosity away...

disableLoaders.gsh

```
:set verbosity QUIET
grouperSession = GrouperSession.startRootSession();
sqlLoaderDefName = AttributeDefNameFinder.findByName("etc:legacy:attribute:legacyGroupType_grouperLoader",
true);
sqlLoaderDefScheduleName = AttributeDefNameFinder.findByName("etc:legacy:attribute:
legacyAttribute_grouperLoaderQuartzCron", true);
ldapLoaderDefName = AttributeDefNameFinder.findByName("etc:attribute:loaderLdap:grouperLoaderLdap", true);
ldapLoaderDefScheduleName = AttributeDefNameFinder.findByName("etc:attribute:loaderLdap:
grouperLoaderLdapQuartzCron", true);
result = new StringBuilder();
```

```
result.append("\n\n############  RESTORE OLD SCHEDULES, BOUNCE GROUPER DAEMONS AFTERWARDS
#############\n\ngrouperSession = GrouperSession.startRootSession();\n");
result.append("sqlLoaderDefName = AttributeDefNameFinder.findByName(\"etc:legacy:attribute:
legacyGroupType_grouperLoader\", true);\n");
result.append("sqlLoaderDefScheduleName = AttributeDefNameFinder.findByName(\"etc:legacy:attribute:
legacyAttribute_grouperLoaderQuartzCron\", true);\n");
result.append("ldapLoaderDefName = AttributeDefNameFinder.findByName(\"etc:attribute:loaderLdap:
grouperLoaderLdap\", true);\n");
result.append("ldapLoaderDefScheduleName = AttributeDefNameFinder.findByName(\"etc:attribute:loaderLdap:
grouperLoaderLdapQuartzCron\", true);\n");
attributeAssigns = GrouperDAOFactory.getFactory().getAttributeAssign().findAttributeAssignments
(AttributeAssignType.group, null, sqlLoaderDefName.getId(), null, null, null, null, null, true, false);
for (AttributeAssign attributeAssign : attributeAssigns) {result.append("group = GroupFinder.findByName
(grouperSession, \"" + attributeAssign.getOwnerGroup().getName() + "\");\nattributeAssignOnAssign = group.
getAttributeDelegate().retrieveAssignment(null, sqlLoaderDefName, false, false);\nattributeAssignOnAssign.
getAttributeValueDelegate().assignValueString(\"" + sqlLoaderDefScheduleName.getName() + "\", \"" +
attributeAssign.getAttributeValueDelegate().retrieveValueString(sqlLoaderDefScheduleName.getName()) + "\");
\n"); }
attributeAssigns = GrouperDAOFactory.getFactory().getAttributeAssign().findAttributeAssignments
(AttributeAssignType.group, null, ldapLoaderDefName.getId(), null, null, null, null, null, true, false);
for (AttributeAssign attributeAssign : attributeAssigns) {result.append("group = GroupFinder.findByName
(grouperSession, \"" + attributeAssign.getOwnerGroup().getName() + "\");\nattributeAssignOnAssign = group.
getAttributeDelegate().retrieveAssignment(null, ldapLoaderDefName, false, false);\nattributeAssignOnAssign.
getAttributeValueDelegate().assignValueString(\"" + ldapLoaderDefScheduleName.getName() + "\", \"" +
attributeAssign.getAttributeValueDelegate().retrieveValueString(ldapLoaderDefScheduleName.getName()) + "\");
\n"); }
result.append("\n\n############  DISABLE ALL SCHEDULES, BOUNCE GROUPER DAEMONS AFTERWARDS
#############\n\ngrouperSession = GrouperSession.startRootSession();\n");
result.append("sqlLoaderDefName = AttributeDefNameFinder.findByName(\"etc:legacy:attribute:
legacyGroupType_grouperLoader\", true);\n");
result.append("sqlLoaderDefScheduleName = AttributeDefNameFinder.findByName(\"etc:legacy:attribute:
legacyAttribute_grouperLoaderQuartzCron\", true);\n");
result.append("ldapLoaderDefName = AttributeDefNameFinder.findByName(\"etc:attribute:loaderLdap:
grouperLoaderLdap\", true);\n");
result.append("ldapLoaderDefScheduleName = AttributeDefNameFinder.findByName(\"etc:attribute:loaderLdap:
grouperLoaderLdapQuartzCron\", true);\n");
attributeAssigns = GrouperDAOFactory.getFactory().getAttributeAssign().findAttributeAssignments
(AttributeAssignType.group, null, sqlLoaderDefName.getId(), null, null, null, null, null, true, false);
for (AttributeAssign attributeAssign : attributeAssigns) {result.append("group = GroupFinder.findByName
(grouperSession, \"" + attributeAssign.getOwnerGroup().getName() + "\");\nattributeAssignOnAssign = group.
getAttributeDelegate().retrieveAssignment(null, sqlLoaderDefName, false, false);\nattributeAssignOnAssign.
getAttributeValueDelegate().assignValueString(\"" + sqlLoaderDefScheduleName.getName() + "\", \"0 0 0 1 1 ?
3000\");\n"); }
attributeAssigns = GrouperDAOFactory.getFactory().getAttributeAssign().findAttributeAssignments
(AttributeAssignType.group, null, ldapLoaderDefName.getId(), null, null, null, null, null, true, false);
for (AttributeAssign attributeAssign : attributeAssigns) {result.append("group = GroupFinder.findByName
(grouperSession, \"" + attributeAssign.getOwnerGroup().getName() + "\");\nattributeAssignOnAssign = group.
getAttributeDelegate().retrieveAssignment(null, ldapLoaderDefName, false, false);\nattributeAssignOnAssign.
getAttributeValueDelegate().assignValueString(\"" + ldapLoaderDefScheduleName.getName() + "\", \"0 0 0 1 1 ?
3000\");\n"); }
System.out.println(result);
```

Example: was run against a server with two jobs that are both scheduled to run at "0 0 * * * ?" .

Example of running disableLoaders.gsh

```
[appadmin@i2midev6 bin]$ ./gsh disableLoaders.gsh

############  RESTORE OLD SCHEDULES, BOUNCE GROUPER DAEMONS AFTERWARDS  #############

grouperSession = GrouperSession.startRootSession();
sqlLoaderDefName = AttributeDefNameFinder.findByName("etc:legacy:attribute:legacyGroupType_grouperLoader",
true);
sqlLoaderDefScheduleName = AttributeDefNameFinder.findByName("etc:legacy:attribute:
legacyAttribute_grouperLoaderQuartzCron", true);
ldapLoaderDefName = AttributeDefNameFinder.findByName("etc:attribute:loaderLdap:grouperLoaderLdap", true);
ldapLoaderDefScheduleName = AttributeDefNameFinder.findByName("etc:attribute:loaderLdap:
grouperLoaderLdapQuartzCron", true);
group = GroupFinder.findByName(grouperSession, "nyu_apereo:presenter:allStevens3");
```

```
attributeAssignOnAssign = group.getAttributeDelegate().retrieveAssignment(null, sqlLoaderDefName, false, false);
attributeAssignOnAssign.getAttributeValueDelegate().assignValueString("etc:legacy:attribute:
legacyAttribute_grouperLoaderQuartzCron", "0 0 * * * ?");
group = GroupFinder.findByName(grouperSession, "test:loader:testLdapGroupList");
attributeAssignOnAssign = group.getAttributeDelegate().retrieveAssignment(null, ldapLoaderDefName, false,
false);
attributeAssignOnAssign.getAttributeValueDelegate().assignValueString("etc:attribute:loaderLdap:
grouperLoaderLdapQuartzCron", "0 0 * * * ?");


############  DISABLE ALL SCHEDULES, BOUNCE GROUPER DAEMONS AFTERWARDS  #############

grouperSession = GrouperSession.startRootSession();
sqlLoaderDefName = AttributeDefNameFinder.findByName("etc:legacy:attribute:legacyGroupType_grouperLoader",
true);
sqlLoaderDefScheduleName = AttributeDefNameFinder.findByName("etc:legacy:attribute:
legacyAttribute_grouperLoaderQuartzCron", true);
ldapLoaderDefName = AttributeDefNameFinder.findByName("etc:attribute:loaderLdap:grouperLoaderLdap", true);
ldapLoaderDefScheduleName = AttributeDefNameFinder.findByName("etc:attribute:loaderLdap:
grouperLoaderLdapQuartzCron", true);
group = GroupFinder.findByName(grouperSession, "nyu_apereo:presenter:allStevens3");
attributeAssignOnAssign = group.getAttributeDelegate().retrieveAssignment(null, sqlLoaderDefName, false, false);
attributeAssignOnAssign.getAttributeValueDelegate().assignValueString("etc:legacy:attribute:
legacyAttribute_grouperLoaderQuartzCron", "0 0 0 1 1 ? 3000");
group = GroupFinder.findByName(grouperSession, "test:loader:testLdapGroupList");
attributeAssignOnAssign = group.getAttributeDelegate().retrieveAssignment(null, ldapLoaderDefName, false,
false);
attributeAssignOnAssign.getAttributeValueDelegate().assignValueString("etc:attribute:loaderLdap:
grouperLoaderLdapQuartzCron", "0 0 0 1 1 ? 3000");
```

Rules

In Grouper 2.3 the UI can delete inherited privileges rules.

To delete a rule, find it in the database in grouper_rules_v. Get the attributeAssignId

```
GrouperSession grouperSession = GrouperSession.startRootSession();
AttributeAssign attributeAssign = AttributeAssignFinder.findById("b629bd8170964663be507968752f4f17", true);
attributeAssign.delete();
```

NOTE: You can also use the AttributeAssignFinder.findById(String id, boolean exceptionIfNull) to find attribute assignments from the logs too. Example log "ERROR RuleEngine$3.callback(560) - - Error with daemon on rule: attributeAssignTypeId: 3d6ccb6c5a584f32919682ae154c0523". id="3d6ccb6c5a584f32919682ae154c0523". The returned AttributeAssign object will show you the stem/group that the attribute is attached to. 🙂

Grouper Builtin Messaging

Create queues / topics, assign privileges for Grouper builtin messaging (not activemq, rabbitmq, AWS, etc) (Grouper 2.3+)

```
grouperSession = GrouperSession.startRootSession();

// create objects
GrouperBuiltinMessagingSystem.createQueue("abc");
GrouperBuiltinMessagingSystem.createTopic("def");


// delete objects
GrouperBuiltinMessagingSystem.deleteQueue("abc");
GrouperBuiltinMessagingSystem.deleteTopic("def");

// permissions on objects
GrouperBuiltinMessagingSystem.allowSendToQueue("abc", SubjectTestHelper.SUBJ0);
GrouperBuiltinMessagingSystem.allowSendToTopic("abc", SubjectTestHelper.SUBJ0);
GrouperBuiltinMessagingSystem.allowReceiveFromQueue("abc", SubjectTestHelper.SUBJ0);
GrouperBuiltinMessagingSystem.disallowSendToQueue("abc", SubjectTestHelper.SUBJ0);
```

```
GrouperBuiltinMessagingSystem.disallowSendToTopic("abc", SubjectTestHelper.SUBJ0);
GrouperBuiltinMessagingSystem.disallowReceiveFromQueue("abc", SubjectTestHelper.SUBJ0);

// topics send to queues
GrouperBuiltinMessagingSystem.topicAddSendToQueue("def", "abc");
Collection<String> queues = GrouperBuiltinMessagingSystem.queuesTopicSendsTo("def");
GrouperBuiltinMessagingSystem.topicRemoveSendToQueue("def", "abc");
```

Grouper messaging

Send, receive, acknowledge messages in any message system (Grouper builtin, activeMQ, rabbitmq, AWS, etc) (Grouper 2.3+)

```
//note, or whatever user should be sending the messages
grouperSession = GrouperSession.startRootSession();

//send message to queue
GrouperMessagingEngine.send(new GrouperMessageSendParam().assignGrouperMessageSystemName
(GrouperBuiltinMessagingSystem.BUILTIN_NAME).assignQueueType(GrouperMessageQueueType.queue).
assignQueueOrTopicName("queueName").addMessageBody("Some message body"));


//send message to topic
GrouperMessagingEngine.send(new GrouperMessageSendParam().assignGrouperMessageSystemName
(GrouperBuiltinMessagingSystem.BUILTIN_NAME).assignQueueType(GrouperMessageQueueType.topic).
assignQueueOrTopicName("queueName").addMessageBody("Some message body"));


//receive messages
GrouperMessageReceiveResult grouperMessageReceiveResult = GrouperMessagingEngine.receive(new
GrouperMessageReceiveParam().assignGrouperMessageSystemName(GrouperBuiltinMessagingSystem.BUILTIN_NAME).
assignQueueName(queueName));


Collection<GrouperMessage> grouperMessages = grouperMessageReceiveResult.getGrouperMessages();


//acknowledge message as processed
GrouperMessagingEngine.acknowledge(new GrouperMessageAcknowledgeParam().assignAcknowledgeType
(GrouperMessageAcknowledgeType.mark_as_processed).assignQueueName("abc").addGrouperMessage(grouperMessage).
assignGrouperMessageSystemName(GrouperBuiltinMessagingSystem.BUILTIN_NAME));

//acknowledge message as return to queue (receive next time ask for messages)
GrouperMessagingEngine.acknowledge(new GrouperMessageAcknowledgeParam().assignAcknowledgeType
(GrouperMessageAcknowledgeType.return_to_queue).assignQueueName("abc").addGrouperMessage(grouperMessage).
assignGrouperMessageSystemName(GrouperBuiltinMessagingSystem.BUILTIN_NAME));

//acknowledge message as return to queue (receive after other messages on the queue)
GrouperMessagingEngine.acknowledge(new GrouperMessageAcknowledgeParam().assignAcknowledgeType
(GrouperMessageAcknowledgeType.return_to_end_of_queue).assignQueueName("abc").addGrouperMessage(grouperMessage).
assignGrouperMessageSystemName(GrouperBuiltinMessagingSystem.BUILTIN_NAME));

//acknowledge message send to another queue or topic (e.g. dead letter queue, dlq)
GrouperMessagingEngine.acknowledge(new GrouperMessageAcknowledgeParam().assignAcknowledgeType
(GrouperMessageAcknowledgeType.send_to_another_queue).assignQueueName("abc").addGrouperMessage(grouperMessage).
assignGrouperMessageSystemName(GrouperBuiltinMessagingSystem.BUILTIN_NAME).assignAnotherQueueParam(new
GrouperMessageQueueParam().assignQueueOrTopicName("dlq").assignQueueType(GrouperMessageQueueType.queue));
```

Expression language testing

```
```

```
Set this in log4j.properties


log4j.logger.edu.internet2.middleware.grouper.util.GrouperUtil = DEBUG


Run GSH:


gsh 0% GrouperSession grouperSession = GrouperSession.startRootSession();
gsh 1% Group group = GroupFinder.findByName(grouperSession, "apps:loader");
gsh 2% Map variableMap =  new HashMap();
gsh 3% variableMap.put("theGroup", group);
gsh 4% String result = GrouperUtil.substituteExpressionLanguage("Name: ${theGroup.name}", variableMap);
gsh 5% result
Name: apps:loader

This is the log entry:

2018-06-04 22:32:58,197: [main] DEBUG GrouperUtil.substituteExpressionLanguage(9416) -  - Subsituting EL:
'Name: ${theGroup.name}', and with env vars: theGroup, grouperUtil with result: 'Name: apps:loader'
```

Example of attribute value assign to group and SQL query

```
gsh 0% GrouperSession grouperSession = GrouperSession.startRootSession();
gsh 1% AttributeDef attributeDef = new AttributeDefSave(grouperSession).assignName("test:testAttribute:
someAttrDef").assignCreateParentStemsIfNotExist(true).assignToGroup(true).assignAttributeDefType
(AttributeDefType.attr).assignMultiAssignable(false).assignMultiValued(false).assignValueType
(AttributeDefValueType.string).save();
gsh 2% AttributeDefName attributeDefName = new AttributeDefNameSave(grouperSession, attributeDef).assignName
("test:testAttribute:someAttr").assignCreateParentStemsIfNotExist(true).save();
gsh 3% Group group = new GroupSave(grouperSession).assignName("test:testAttribute:group").
assignCreateParentStemsIfNotExist(true).save()
gsh 4% group.getAttributeValueDelegate().assignValueString(attributeDefName.getName(), "someValue");
edu.internet2.middleware.grouper.attr.value.AttributeValueResult: edu.internet2.middleware.grouper.attr.value.
AttributeValueResult@2f08e6d3
gsh 5% HibernateSession.bySqlStatic().select(String.class, "SELECT value_string FROM grouper_aval_asn_group_v
WHERE group_name = 'test:testAttribute:group' AND attribute_def_name_name = 'test:testAttribute:someAttr'");
someValue
gsh 6%
```

Example of finding groups with a certain attribute value

```
GrouperSession grouperSession = GrouperSession.startRootSession();
Set<Group> groups = new GroupFinder().assignNameOfAttributeDefName("bath:provisionClass").assignAttributeValue
("groupmanager-groups")
       .assignPrivileges(AccessPrivilege.ATTRIBUTE_READ_PRIVILEGES).findGroups();
```

Example of finding groups with a certain attribute value on metadata assignments

This is useful for attestation and loader metadata, for example.

```
//groups loaded by a particular group
def attrDef = AttributeDefNameFinder.findByName(GrouperCheckConfig.loaderMetadataStemName() + ":" +
GrouperLoader.ATTRIBUTE_GROUPER_LOADER_METADATA_GROUP_ID, true)
def groups = new GroupFinder().assignIdOfAttributeDefName(attrDef.id).assignAttributeValuesOnAssignment
(GrouperUtil.toSetObjectType("3779b51223804784b4a02ee238b73079")).findGroups()
```

```
//    OR, without the extra attributeDef, look up the name directly in GroupFinder
def groups = new GroupFinder().assignNameOfAttributeDefName("etc:attribute:loaderMetadata:
grouperLoaderMetadataGroupId").assignAttributeValuesOnAssignment(GrouperUtil.toSetObjectType
("3779b51223804784b4a02ee238b73079")).findGroups()


//attestations that are due
import edu.internet2.middleware.grouper.app.attestation.GrouperAttestationJob
def groups = new GroupFinder().assignIdOfAttributeDefName(GrouperAttestationJob.
retrieveAttributeDefNameCalculatedDaysLeft().id).assignAttributeValuesOnAssignment(GrouperUtil.toSetObjectType
("0")).findGroups()
```

Example of finding provisioning targets for PSPNG

Example of finding provisioning targets for PSPNG

```
gsh 0% HibernateSession.bySqlStatic().listSelect(String.class, "SELECT DISTINCT gaaa.value_string FROM
grouper_attribute_assign_value gaaa, grouper_attribute_assign gaa, grouper_attribute_def_name gadn WHERE gaaa.
attribute_assign_id = gaa.id AND gaa.attribute_def_name_id = gadn.id AND gadn.extension IN ('provision_to',
'do_not_provision_to')", null, null);
java.util.ArrayList: [ad, ldap]
```

Example of finding which groups are provisioned to a certain target (from daemon where PSPNG is installed)

```
provisioner_name="xyz"; // Whatever your provisioner is called in grouper_loader.properties
gs=GrouperSession.startRootSession();
provisioner=edu.internet2.middleware.grouper.pspng.ProvisionerFactory.createProvisioner(provisioner_name,false);
provisioner.getAllGroupsForProvisioner();
```

Long hand example of which groups are provisioned to a certain target

```
    String provisionTarget = "ad";
    GrouperSession grouperSession = GrouperSession.startRootSession();

    Set stemsToProvisionToSet = HibernateSession.byHqlStatic().createQuery("select s from Stem s,
AttributeAssign aa, AttributeDefName adn, AttributeAssignValue aav where s.id = aa.ownerStemId and aav.
attributeAssignId = aa.id and aa.attributeDefNameId = adn.id and aa.attributeAssignTypeDb = 'stem' and aa.
enabledDb = 'T' and adn.extensionDb = 'provision_to' and aav.valueString = '" + provisionTarget + "'").listSet
(Stem.class);
    for (Object stemObject : stemsToProvisionToSet) { Stem stem = (Stem)stemObject; System.out.println
("provision_to assigned to stem: " + stem.getName());  }
    Set stemsToNotProvisionToSet = HibernateSession.byHqlStatic().createQuery("select s from Stem s,
AttributeAssign aa, AttributeDefName adn, AttributeAssignValue aav where s.id = aa.ownerStemId and aav.
attributeAssignId = aa.id and aa.attributeDefNameId = adn.id and aa.attributeAssignTypeDb = 'stem' and aa.
enabledDb = 'T' and adn.extensionDb = 'do_not_provision_to' and aav.valueString = '" + provisionTarget + "'").
listSet(Stem.class);
    for (Object stemObject : stemsToNotProvisionToSet) { Stem stem = (Stem)stemObject; System.out.println
("do_not_provision_to assigned to stem: " + stem.getName());  }
    Set groupsToProvisionToSet = HibernateSession.byHqlStatic().createQuery("select g from Group g,
AttributeAssign aa, AttributeDefName adn, AttributeAssignValue aav where g.id = aa.ownerGroupId and aav.
attributeAssignId = aa.id and aa.attributeDefNameId = adn.id and aa.attributeAssignTypeDb = 'group' and aa.
enabledDb = 'T' and adn.extensionDb = 'provision_to' and aav.valueString = '" + provisionTarget + "'").listSet
(Stem.class);
    for (Object groupObject : groupsToProvisionToSet) { Group group = (Group)groupObject; System.out.println
("provision_to assigned to group: " + group.getName());  }
    Set groupsToNotProvisionToSet = HibernateSession.byHqlStatic().createQuery("select g from Group g,
AttributeAssign aa, AttributeDefName adn, AttributeAssignValue aav where g.id = aa.ownerGroupId and aav.
attributeAssignId = aa.id and aa.attributeDefNameId = adn.id and aa.attributeAssignTypeDb = 'group' and aa.
enabledDb = 'T' and adn.extensionDb = 'do_not_provision_to' and aav.valueString = '" + provisionTarget + "'").
listSet(Stem.class);
    for (Object groupObject : groupsToNotProvisionToSet) { Group group = (Group)groupObject; System.out.println
("do_not_provision_to assigned to group: " + group.getName());  }
```

```
    Set allGroups = new LinkedHashSet();
    Set allGroupsToProvision = new TreeSet();
    allGroupsToProvision.addAll(groupsToProvisionToSet);

    Set stemNamesToNotProvisionTo = new HashSet();
    Set stemNamesToProvisionTo = new HashSet();

    for (Object stemToProvision : stemsToProvisionToSet) { stemNamesToProvisionTo.add(((Stem)stemToProvision).
getName()); }
    for (Object stemNotToProvision : stemsToNotProvisionToSet) { stemNamesToNotProvisionTo.add(((Stem)
stemNotToProvision).getName()); }

    for (Object stemToProvision : stemsToProvisionToSet) { allGroups.addAll(((Stem)stemToProvision).
getChildGroups(edu.internet2.middleware.grouper.Stem.Scope.SUB)); }

    Map groupToPaths = new HashMap();
    for (Object groupObject : allGroups) { Group group = (Group)groupObject; if (allGroupsToProvision.contains
(group)) {continue;} if (groupsToNotProvisionToSet.contains(group)) {continue;} List paths = new ArrayList();
groupToPaths.put(group, paths); String currentName = group.getName(); paths.add(currentName);  while(true) {
currentName = GrouperUtil.parentStemNameFromName(currentName);  if (GrouperUtil.isBlank(currentName)) {break;}
paths.add(currentName);  }   }

    for (Object groupObject : groupToPaths.keySet()) {Group group = (Group)groupObject; List paths = (List)
groupToPaths.get(group); for (Object pathObject : paths) { String path = (String)pathObject; if
(stemNamesToProvisionTo.contains(path)) { allGroupsToProvision.add(group); break; } if
(stemNamesToNotProvisionTo.contains(path)) { break; } } }

    for (Object groupObject : allGroupsToProvision) { Group group = (Group)groupObject; System.out.println
("configured to provision to: " + provisionTarget + ": " + group.getName()); }
```

Include a common GSH file

If this is the file in the container:

```
def  addNewGroup(pathName, dispName,groupType, gsh_builtin_gshTemplateOutput,grouperSession ) {
    //Verify if allowgroup exist or not and provide appropriate messages
   groupVerify = GroupFinder.findByName(grouperSession, pathName,false);
   if (groupVerify != null)
        gsh_builtin_gshTemplateOutput.addOutputLine( pathName + " group already exists!");
    else
    {
        productGroup = new GroupSave(grouperSession).assignCreateParentStemsIfNotExist(true).assignName
(pathName).assignDisplayName(dispName).save();
        gsh_builtin_gshTemplateOutput.addOutputLine("Created "+ groupType +" group : "  +pathName);
        gsh_builtin_gshTemplateOutput.addOutputLine("Added Group types: "+groupType+" for group : " + pathName);
    }

GdgTypeGroupSave gdgTypeGroupSave = new GdgTypeGroupSave().assignGroup(productGroup).assignType(groupType);
gdgTypeGroupSave.save();
return productGroup;

}
```

Include that in a GSH template or other script

```
GroovyShell shell = new GroovyShell();
def external = shell.parse(new File('/opt/grouper/gsh/commonFunction.gsh'));

newGroup=external.addNewGroup
```

Stem move

try this:

```
GrouperSession.startRootSession();
stemFrom = StemFinder.findByName(grouperSession, "a:b", true);
stemTo = StemFinder.findByName(grouperSession, "a:c", true);
new edu.internet2.middleware.grouper.StemMove(stemFrom, stemTo).assignAlternateName(false).save();
```

Check health of database connection or run a query

(in 2.4.0 api patch 93+)

```
gcDbAccess = new edu.internet2.middleware.grouperClient.jdbc.GcDbAccess();
gcDbAccess.connectionName("warehouse").sql("select count(1) from grouper_groups").select(int.class);
```

Set password using Grouper built-in authentication

```
v2.5.29+
new GrouperPasswordSave().assignApplication(GrouperPassword.Application.UI).assignUsername("username").
assignPassword("password").save();
```

Remove all group / folder privs for a user. Remove user from groups which have a privilege on another object

```
import java.util.List;
import java.util.Set;

import edu.internet2.middleware.grouper.FieldType;
import edu.internet2.middleware.grouper.Group;
import edu.internet2.middleware.grouper.GroupFinder;
import edu.internet2.middleware.grouper.GroupSave;
import edu.internet2.middleware.grouper.GrouperSession;
import edu.internet2.middleware.grouper.Member;
import edu.internet2.middleware.grouper.Membership;
import edu.internet2.middleware.grouper.MembershipFinder;
import edu.internet2.middleware.grouper.Stem;
import edu.internet2.middleware.grouper.StemSave;
import edu.internet2.middleware.grouper.SubjectFinder;
import edu.internet2.middleware.grouper.membership.MembershipType;
import edu.internet2.middleware.grouper.privs.AccessPrivilege;
import edu.internet2.middleware.grouper.privs.NamingPrivilege;
import edu.internet2.middleware.grouper.util.GrouperUtil;
import edu.internet2.middleware.grouperClient.jdbc.GcDbAccess;
import edu.internet2.middleware.subject.Subject;

//public class Test36revokePrivs {
//
//  public static void main(String[] args) {

    GrouperSession grouperSession = GrouperSession.startRootSession();

    String subjectId = "test.subject.0";
    String subjectSourceId = "jdbc";

    Subject subject = SubjectFinder.findByIdAndSource(subjectId, subjectSourceId, true);

    Group group1 = new GroupSave().assignName("test:test1").assignCreateParentStemsIfNotExist(true).save();
    Group group2 = new GroupSave().assignName("test:test2").assignCreateParentStemsIfNotExist(true).save();


    group1.delete();
```

```
    group2.delete();

    group1 = new GroupSave().assignName("test:test1").assignCreateParentStemsIfNotExist(true).save();
    group2 = new GroupSave().assignName("test:test2").assignCreateParentStemsIfNotExist(true).save();

    group1.grantPriv(subject, AccessPrivilege.READ, false);
    group1.grantPriv(subject, AccessPrivilege.UPDATE, false);
    group2.grantPriv(subject, AccessPrivilege.ADMIN, false);

    group2.addMember(subject);
    group1.addMember(subject);
    group2.grantPriv(group1.toSubject(), AccessPrivilege.READ, false);

    Stem stem1 = new StemSave().assignName("test1").assignCreateParentStemsIfNotExist(true).save();
    stem1.grantPriv(subject, NamingPrivilege.CREATE, false);
    stem1.grantPriv(subject, NamingPrivilege.STEM_ATTR_READ, false);
    Stem stem2 = new StemSave().assignName("test2").assignCreateParentStemsIfNotExist(true).save();
    stem2.grantPriv(subject, NamingPrivilege.STEM_ADMIN, false);

    List<String> groupNames = new GcDbAccess().sql("select gg.name from grouper_groups gg where exists (" +
      " select 1 from grouper_memberships gmem, grouper_fields gf, grouper_members gm " +
      " where gmem.field_id = gf.id and gf.type in ('naming', 'access', 'attributeDef') " +
      " and gmem.member_id = gm.id and gm.subject_id = gg.id and gm.subject_source = 'g:gsa') " +
      " and exists (select 1 from grouper_memberships gmem, grouper_fields gf, grouper_members gm " +
      " where gmem.field_id = gf.id and gf.name = 'members' " +
      " and gm.subject_id = ? " +
      " and gmem.member_id = gm.id and gm.subject_source = '" + subjectSourceId + "')").addBindVar(subjectId).
selectList(String.class);

    for (String groupName : GrouperUtil.nonNull(groupNames)) {
      Group group = GroupFinder.findByName(groupName, true);
      group.deleteMember(subject, false);
      System.out.println("Deleted membership from group: " + group.getName() + ", since group has privilege on
another object");
    }

    Set<Object[]> membershipsOwnersMembers = new MembershipFinder().addSubject(subject).assignFieldType
(FieldType.ACCESS).
      assignMembershipType(MembershipType.IMMEDIATE).findMembershipResult().getMembershipsOwnersMembers();

    for (Object[] membershipOwnerMember : GrouperUtil.nonNull(membershipsOwnersMembers)) {
      Membership membership = (Membership)membershipOwnerMember[0];
      Group group = (Group)membershipOwnerMember[1];
      Member member = (Member)membershipOwnerMember[2];
      group.revokePriv(member.getSubject(), AccessPrivilege.listToPriv(membership.getField().getName() ));
      System.out.println("Deleted priv from group: " + group.getName() + ": " + membership.getField().
getName());
    }

    membershipsOwnersMembers = new MembershipFinder().addSubject(subject).assignFieldType(FieldType.NAMING).
        assignMembershipType(MembershipType.IMMEDIATE).findMembershipResult().getMembershipsOwnersMembers();

    for (Object[] membershipOwnerMember : GrouperUtil.nonNull(membershipsOwnersMembers)) {
      Membership membership = (Membership)membershipOwnerMember[0];
      Stem stem = (Stem)membershipOwnerMember[1];
      Member member = (Member)membershipOwnerMember[2];
      stem.revokePriv(member.getSubject(), NamingPrivilege.listToPriv(membership.getField().getName() ));
      System.out.println("Deleted priv from folder: " + stem.getName() + ": " + membership.getField().
getName());
    }
//
//  }
//}
//Deleted membership from group: test:test1, since group has privilege on another object
//Deleted priv from group: test:test1: readers
//Deleted priv from group: test:test1: updaters
//Deleted priv from group: test:test2: admins
//Deleted priv from folder: test2: stemAdmins
//Deleted priv from folder: test1: creators
//Deleted priv from folder: test1: stemAttrReaders
```

Delete all members from a subject source

Note that it is not normal to remove entries from the grouper_members table, which is what this example is doing. But, for example, this may be useful if you temporarily added a subject source that you need to get rid of completely. If you need to instead merge members from one source to another, there's another option here - Member change subject

Also note that this doesn't handle all the foreign keys that may exist on members. So tweaks may be needed if some of the members are not able to be deleted.

```
String sourceId = "sourceIdToDelete";
Set<Member> members = HibernateSession.byHqlStatic().createQuery("from Member as m where m.subjectSourceIdDb=:
sourceId").setString("sourceId", sourceId).listSet(Member.class);
System.out.println("Found " + members.size() + " members to delete");

Set<Field> fields = FieldFinder.findAll();
for (Member member : members) {
  for (Field field : fields) {
    for (Membership membership : member.getImmediateMemberships(field)) {
      System.out.println("Deleting membership with id=" + membership.getImmediateMembershipId());
      membership.delete();
    }
  }

  try {
    HibernateSession.byObjectStatic().delete(member);
    System.out.println("Deleted " + member.getSubjectId());
  } catch (Exception e) {
    e.printStackTrace();
  }
}
```