

Grouper export to a GSH script

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

Grouper can export a folder or the entire registry as a GSH script. It is important to do a backup or verify in a test env first. In order to run the script you need a Grouper v2.2.2+ patched env. Note, you should capture the output of running the GSH script so you can review it for errors.

Performance on demo server (local mysql database)

- Exporting 6000 records to GSH takes 26 seconds
- The GSH script ran 6000 records 1min 6sec with no changes
- If there were 6000 changes (new registry) it took 6 minutes.

The script is idempotent (run it twice, its ok). It will not delete what is there in general though it will delete attribute assignments on exported assignments. It is assumed that the general configuration of the source environment is equivalent to the destination environment. e.g. if name is not a required external subject attribute in the source env, then it shouldnt be in the destination either.

It will not export

- audits
- point in time data
- daemon logs
- change log entries
- if you are importing and the objects already exist, it will not sync up the object ID uuid's. In some cases will be use different uuid even if the object doesnt exist
- membership fields are not imported (note, privileges are, but if you are using a membership list that is not "members" (not common), that will be skipped
- if you export a folder, and some necessary objects do not exist in the target (e.g. attribute definitions), they will be reported as an error and skipped
- note: international characters have been tested on the demo server and works there though your mileage may vary, be careful here (test it)
- unresolvable subjects will be exported but will throw an error on import (maybe they shouldnt be exported)

Start an export

```
grouperSession = GrouperSession.startRootSession();
scriptName = "c:/temp/script.gsh";
new File(scriptName).delete();
new XmlExportGshScript().assignStemName(":").assignFileNameToWriteTo(scriptName).exportGsh();

or

new XmlExportGshScript().assignStemName("some:folder").assignFileNameToWriteTo(scriptName).exportGsh();

or

new XmlExportGshScript().addObjectName("some:group").addObjectName("another:group").assignFileNameToWriteTo(
scriptName).exportGsh();
```

The end of the script will print a summary of changes that occurred:

```
Script complete: total objects: 6222, expected approx total: 6327, changes: 133, known errors (view output for
full list): 36
```

Example script

```
GrouperSession grouperSession = GrouperSession.startRootSession();
long gshTotalObjectCount = 0L;
long gshTotalChangeCount = 0L;
long gshTotalErrorCount = 0L;
StemSave stemSave = new StemSave(grouperSession).assignName(":").assignCreateParentStemsIfNotExist(true).
assignDisplayName(":");
stem = stemSave.save();
gshTotalObjectCount++;
```

```

System.out.println("Done with folders, objects: " + gshTotalObjectCount + ", expected approx total: 6,327,
changes: " + gshTotalChangeCount + ", known errors (view output for full list): " + gshTotalErrorCount);
GroupSave groupSave = new GroupSave(grouperSession).assignName("aStem:library").
assignCreateParentStemsIfNotExist(true).assignDescription("access to the library application").assignDisplayName
("aStem:library").assignTypeOfGroup(TypeOfGroup.group);
Group group = groupSave.save();
gshTotalObjectCount++;
System.out.println("Done with groups, objects: " + gshTotalObjectCount + ", expected approx total: 6,327,
changes: " + gshTotalChangeCount + ", known errors (view output for full list): " + gshTotalErrorCount);
Group ownerGroup = GroupFinder.findByName(grouperSession, "users:penn:hayese:testGroup2", false);
Group leftFactorGroup = GroupFinder.findByName(grouperSession, "users:penn:hayese:testGroup2_systemOfRecord",
false);
Group rightFactorGroup = GroupFinder.findByName(grouperSession, "community:students", false);
CompositeType compositeType = CompositeType.INTERSECTION;
if (ownerGroup != null) { if (leftFactorGroup != null) { if (rightFactorGroup != null) { CompositeSave
compositeSave = new CompositeSave(grouperSession).assignOwnerGroup(ownerGroup).assignCompositeType
(compositeType).assignLeftFactorGroup(leftFactorGroup).assignRightFactorGroup(rightFactorGroup);
gshTotalObjectCount++; Composite composite = compositeSave.save(); if (compositeSave.getSaveResultType() !=
SaveResultType.NO_CHANGE) {System.out.println("Made change for composite: " + composite.toString());
gshTotalChangeCount++;} } else { System.out.println("ERROR: cant find rightFactorGroup: 'community:
students'"); gshTotalErrorCount++; } } else { System.out.println("ERROR: cant find leftFactorGroup: 'users:
penn:hayese:testGroup2_systemOfRecord'"); gshTotalErrorCount++; } } else { System.out.println("ERROR: cant
find overallGroup: 'users:penn:hayese:testGroup2'"); gshTotalErrorCount++; }
System.out.println("Done with composites, objects: " + gshTotalObjectCount + ", expected approx total: 6,327,
changes: " + gshTotalChangeCount + ", known errors (view output for full list): " + gshTotalErrorCount);
AttributeDefSave attributeDefSave = new AttributeDefSave(grouperSession).assignName("apps:issueTracker:
issueTrackerServiceDef").assignCreateParentStemsIfNotExist(true).assignToStem(true).assignAttributeDefType
(AttributeDefType.service).assignMultiAssignable(false).assignMultiValued(false).assignValueType
(AttributeDefValueType.marker);
AttributeDef attributeDef = attributeDefSave.save();
gshTotalObjectCount++;
System.out.println("Done with attribute definitions, objects: " + gshTotalObjectCount + ", expected approx
total: 6,327, changes: " + gshTotalChangeCount + ", known errors (view output for full list): " +
gshTotalErrorCount);
ifHasRole = GroupFinder.findByName(grouperSession, "users:duke:shilen:role", false);
thenHasRole = GroupFinder.findByName(grouperSession, "users:duke:shilen:role2", false);
if (ifHasRole != null) { if (thenHasRole != null) { boolean changed = ifHasRole.getRoleInheritanceDelegate().
addRoleToInheritFromThis(thenHasRole); if (changed) { gshTotalChangeCount++; System.out.println("Made change
for role inheritance: " + ifHasRole.getName() + " then has role " + thenHasRole.getName()); } } else { System.
out.println("ERROR: cant find thenHasRole: " + thenHasRole.getName()); gshTotalErrorCount++; } } else { System.
out.println("ERROR: cant find ifHasRole: " + ifHasRole.getName()); gshTotalErrorCount++; }
System.out.println("Done with role hierarchies, objects: " + gshTotalObjectCount + ", expected approx total:
6,327, changes: " + gshTotalChangeCount + ", known errors (view output for full list): " + gshTotalErrorCount);
attributeDef = AttributeDefFinder.findByName("etc:attribute:attrExternalSubjectInvite:
externalSubjectInviteAttrDef", false);
if (attributeDef != null) { int changeCount = attributeDef.getAttributeDefActionDelegate().configureActionList
("assign"); gshTotalObjectCount++; if (changeCount > 0) { gshTotalChangeCount+=changeCount; System.out.println
("Made " + changeCount + " changes for actionList of attributeDef: etc:attribute:attrExternalSubjectInvite:
externalSubjectInviteAttrDef"); } } else { gshTotalErrorCount++; System.out.println("ERROR: cant find
attributeDef: 'etc:attribute:attrExternalSubjectInvite:externalSubjectInviteAttrDef'"); }
System.out.println("Done with attribute actions, objects: " + gshTotalObjectCount + ", expected approx total:
6,327, changes: " + gshTotalChangeCount + ", known errors (view output for full list): " + gshTotalErrorCount);
AttributeDef attributeDef = AttributeDefFinder.findByName("users:penn:teresh:tereshOrgPermission", false);
if (attributeDef != null) { AttributeAssignAction ifHasAction = attributeDef.getAttributeDefActionDelegate().
findAction("admin", false); if (ifHasAction != null) { AttributeAssignAction thenHasAction = attributeDef.
getAttributeDefActionDelegate().findAction("update", false); if (thenHasAction != null) { boolean changed =
ifHasAction.getAttributeAssignActionSetDelegate().addAttributeAssignActionSet(thenHasAction);
gshTotalObjectCount++; if (changed) { gshTotalChangeCount++; System.out.println("Made change for action
inheritance for attributeDef: users:penn:teresh:tereshOrgPermission, if has action: admin then has action:
update"); } } } else { System.out.println("ERROR: cant find thenHasAction: 'update', attributeDefName: 'users:
penn:teresh:tereshOrgPermission'"); gshTotalErrorCount++; } } else { System.out.println("ERROR: cant find
ifHasAction: 'admin', attributeDefName: 'users:penn:teresh:tereshOrgPermission'"); gshTotalErrorCount++; } }
else { System.out.println("ERROR: cant find attributeDef: 'users:penn:teresh:tereshOrgPermission'");
gshTotalErrorCount++; }
System.out.println("Done with attribute action hierarchies, objects: " + gshTotalObjectCount + ", expected
approx total: 6,327, changes: " + gshTotalChangeCount + ", known errors (view output for full list): " +
gshTotalErrorCount);
Subject subject = SubjectFinder.findByIdAndSource("dc586839e8294ac085c004083a5a88c6", "g:gsa", false);
Privilege privilege = Privilege.listToPriv("stemmers", false);
Stem stem = StemFinder.findByName(grouperSession, "users:penn2:selznick", false);
if (privilege != null) { if (subject != null) { if (stem != null) { boolean changed = stem.grantPriv(subject,

```

```

privilege, false); gshTotalObjectCount++; if (changed) { gshTotalChangeCount++; System.out.println("Made
change for stem privilege: " + stem.getName() + ", privilege: " + privilege + ", subject: " + GrouperUtil.
subjectToString(subject)); } } else { gshTotalErrorCount++; System.out.println("ERROR: cant find stem: 'users:
penn2:selznick'"); } } else { gshTotalErrorCount++; System.out.println("ERROR: cant find subject: 'g:gsa' -->
'dc586839e8294ac085c004083a5a88c6'"); } }
Subject subject = SubjectFinder.findByIdAndSource("03e2103c9ba34bc68fbc7ac91ed0bca8", "g:gsa", false);
Privilege privilege = null;
Group group = GroupFinder.findByName(grouperSession, "users:penn:levinm:MSGGroup_includes", false);
if (subject != null) { if (group != null) { boolean changed = group.addOrEditMember(subject, false, true, null,
null, false); gshTotalObjectCount++; if (changed) { gshTotalChangeCount++; System.out.println("Made
change for group membership: " + group.getName() + ", field: members, subject: " + GrouperUtil.subjectToString
(subject)); } } else { gshTotalErrorCount++; System.out.println("ERROR: cant find group: 'users:penn:levinm:
MSGGroup_includes'"); } } else { gshTotalErrorCount++; System.out.println("ERROR: cant find subject: 'g:gsa' --
> '03e2103c9ba34bc68fbc7ac91ed0bca8'"); }
System.out.println("Done with memberships and privileges, objects: " + gshTotalObjectCount + ", expected approx
total: 6,327, changes: " + gshTotalChangeCount + ", known errors (view output for full list): " +
gshTotalErrorCount);
AttributeDef attributeDef = AttributeDefFinder.findByName("apps:issueTracker:issueTrackerServiceDef", false);
if (attributeDef != null) { AttributeDefNameSave attributeDefNameSave = new AttributeDefNameSave
(grouperSession, attributeDef).assignName("apps:issueTracker:issueTracker").assignCreateParentStemsIfNotExist
(true).assignDisplayName("apps:issueTracker:issueTracker"); AttributeDefName attributeDefName =
attributeDefNameSave.save(); gshTotalObjectCount++; if (attributeDefNameSave.getSaveResultType() !=
SaveResultType.NO_CHANGE) { gshTotalChangeCount++; System.out.println("Made change for attributeDefName: " +
attributeDefName.getName()); } } else { gshTotalErrorCount++; System.out.println("ERROR: cant find
attributeDef: 'apps:issueTracker:issueTrackerServiceDef'"); }
System.out.println("Done with attribute names, objects: " + gshTotalObjectCount + ", expected approx total:
6,327, changes: " + gshTotalChangeCount + ", known errors (view output for full list): " + gshTotalErrorCount);
AttributeDefName ifHasAttributeDefName = AttributeDefNameFinder.findByName("users:penn:mchyzer:all", false);
AttributeDefName thenHasAttributeDefName = AttributeDefNameFinder.findByName("users:penn:mchyzer:engineering",
false);
if (ifHasAttributeDefName != null) { if (thenHasAttributeDefName != null) { boolean changed =
ifHasAttributeDefName.getAttributeDefNameSetDelegate().addToAttributeDefNameSet(thenHasAttributeDefName);
gshTotalObjectCount++; if (changed) { gshTotalChangeCount++; System.out.println("Made change for
attributeDefName inheritance: " + ifHasAttributeDefName.getName() + " implies " + thenHasAttributeDefName.
getName()); } } else { gshTotalErrorCount++; System.out.println("ERROR: cant find thenHasAttributeDefName:
'users:penn:mchyzer:engineering'"); } } else { gshTotalErrorCount++; System.out.println("ERROR: cant find
ifHasAttributeDefName: 'users:penn:mchyzer:all'"); }
System.out.println("Done with attribute name hierarchies, objects: " + gshTotalObjectCount + ", expected approx
total: 6,327, changes: " + gshTotalChangeCount + ", known errors (view output for full list): " +
gshTotalErrorCount);
AttributeDef = AttributeDefFinder.findByName("etc:attribute:attrExternalSubjectInvite:
externalSubjectInviteAttrDef", false);
AttributeDefScopeType = AttributeDefScopeType.valueOfIgnoreCase("nameEquals", true);
if (attributeDef != null) { if (attributeDefScopeType != null) { gshTotalObjectCount++; if (attributeDef.
getAttributeDefScopeDelegate().retrieveAttributeDefScope(attributeDefScopeType, "\"etc:attribute:
attrExternalSubjectInvite:externalSubjectInvite\"", "null") != null) { gshTotalChangeCount++; attributeDef.
getAttributeDefScopeDelegate().assignScope(attributeDefScopeType, "\"etc:attribute:attrExternalSubjectInvite:
externalSubjectInvite\"", "null"); System.out.println("Made change for attributeDefScope on attributeDef: etc:
attribute:attrExternalSubjectInvite:externalSubjectInviteAttrDef, etc:attribute:attrExternalSubjectInvite:
externalSubjectInvite, null"); } } else { gshTotalErrorCount++; System.out.println("ERROR: cant find
attributeDefScopeType: 'nameEquals'"); } } else { gshTotalErrorCount++; System.out.println("ERROR: cant find
attributeDef: 'etc:attribute:attrExternalSubjectInvite:externalSubjectInviteAttrDef'"); }
System.out.println("Done with attribute definition scopes, objects: " + gshTotalObjectCount + ", expected
approx total: 6,327, changes: " + gshTotalChangeCount + ", known errors (view output for full list): " +
gshTotalErrorCount);
Set attributeAssignIdsAlreadyUsed = new HashSet();
boolean problemWithAttributeAssign = false;
AttributeAssignSave attributeAssignSave = new AttributeAssignSave(grouperSession).
assignAttributeAssignIdsToNotUse(attributeAssignIdsAlreadyUsed).assignPrintChangesToSystemOut(true);
attributeAssignSave.assignAttributeAssignType(AttributeAssignType.imm_mem);
AttributeDefName attributeDefName = AttributeDefNameFinder.findByName("etc:attribute:userData:grouperUserData",
false);
if (attributeDefName == null) { gshTotalErrorCount++; System.out.println("Error: cant find attributeDefName:
etc:attribute:userData:grouperUserData"); problemWithAttributeAssign = true; }
attributeAssignSave.assignAttributeDefName(attributeDefName);
attributeAssignSave.assignPutAttributeAssignIdsToNotUseSet(true);
Group ownerGroup = GroupFinder.findByName(grouperSession, "etc:grouperUi:grouperUiUserData", false);
if (ownerGroup == null) { gshTotalErrorCount++; System.out.println("Error: cant find group: etc:grouperUi:
grouperUiUserData"); problemWithAttributeAssign = true; }
attributeAssignSave.assignOwnerGroup(ownerGroup);
Subject ownerSubject = SubjectFinder.findByIdAndSource("5557e047c3214621819bd21c3b91d763", "grouperExternal",

```

```

false);
if (ownerSubject == null) { gshTotalErrorCount++; System.out.println("Error: cant find subject:
grouperExternal: 5557e047c3214621819bd21c3b91d763"); problemWithAttributeAssign = true; }
if (ownerSubject != null) { Member ownerMember = MemberFinder.findBySubject(grouperSession, ownerSubject,
true); attributeAssignSave.assignOwnerMember(ownerMember); }
AttributeAssignSave attributeAssignOnAssignSave = new AttributeAssignSave(grouperSession).
assignAttributeAssignIdsToNotUse(attributeAssignIdsAlreadyUsed).assignPrintChangesToSystemOut(true);
attributeAssignOnAssignSave.assignAttributeAssignType(AttributeAssignType.imm_mem_asgn);
AttributeDefName attributeDefName = AttributeDefNameFinder.findByName("etc:attribute:userData:
grouperUserDataRecentGroups", false);
if (attributeDefName == null) { gshTotalErrorCount++; System.out.println("Error: cant find attributeDefName:
etc:attribute:userData:grouperUserDataRecentGroups"); problemWithAttributeAssign = true; }
attributeAssignOnAssignSave.assignAttributeDefName(attributeDefName);
attributeAssignOnAssignSave.assignPutAttributeAssignIdsToNotUseSet(true);
attributeAssignOnAssignSave.addValue("{\"list\": [{\"timestamp\":1453992092662,\"uuid\": \"
01d1ec77e56f4f63933b95887d67de30\"}, {\"timestamp\":1453991965490,\"uuid\": \"8a860d07dadb46b5bcc65544a6e89350\"},
{\"timestamp\":1447190902786,\"uuid\": \"9d25e467-b127-4229-880b-ba65d0506c02\"}]}");
attributeAssignSave.addAttributeAssignOnThisAssignment(attributeAssignOnAssignSave);
gshTotalObjectCount += 3;
if (!problemWithAttributeAssign) { AttributeAssign attributeAssign = attributeAssignSave.save(); if
(attributeAssignSave.getChangesCount() > 0) { gshTotalChangeCount+=attributeAssignSave.getChangesCount();
System.out.println("Made " + attributeAssignSave.getChangesCount() + " changes for attribute assign: " +
attributeAssign.toString()); } }
System.out.println("Script complete: total objects, objects: " + gshTotalObjectCount + ", expected approx
total: 6,327, changes: " + gshTotalChangeCount + ", known errors (view output for full list): " +
gshTotalErrorCount);

```