

Grouper - Loader LDAP

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

[Main Grouper Loader page](#)

This page documents using the Grouper Loader to load a group from LDAP. This is available in Grouper v2.1 and later

- [Grouper Loader LDAP bulk lookup by filter \(with cache\)](#)
- [Grouper Loader LDAP dereference example convert DN to RDN value](#)
- [Grouper Loader LDAP examples](#)
- [Grouper Loader LDAP simple from hasMember](#)
- [Grouper Loader LDAP simple lookup by DN \(no cache\)](#)
- [Grouper Loader LDAP simple lookup by DN \(with cache\)](#)
- [Grouper Loader LDAP simple lookup by filter \(no cache\)](#)
- [Grouper Loader LDAP simple lookup by filter \(with cache\)](#)

Grouper Loader LDAP configuration

The Grouper Loader LDAP configuration is done through the [attribute framework](#). You can assign the grouperLoaderLdap attribute on a group, and the configuration attributes on that assignment. Note, these attributes are in the attribute root stem name (default "etc:attribute"), in a subfolder named "loaderLdap". By default only Grouper admins can assign or edit these attributes, though an admin could delegate that permission to someone else. Be very careful of the security implications (they could run any LDAP filter to load their group, which could be sensitive data). Note, all LDAP jobs are scheduled as crons. These attributes do not necessarily automatically exist in a new Grouper installation, you could manually create them, or they are automatically created on Grouper startup if they don't exist if the grouper.properties setting: grouper.attribute.loader.autoconfigure is set to true (defaults to true).

Attribute system name	Attribute display name	Required?	Description	Assignable to	Value type	Example value
grouperLoaderLdap	Grouper Loader LDAP	required	This is the marker attribute that you assign to a group to mark is as a grouper loader ldap group	Groups	None	
grouperLoaderLdapType	Grouper Loader LDAP type	required	Like the SQL loader, this holds the type of job from the GrouperLoaderType enum, currently the only valid values are LDAP_SIMPLE, LDAP_GROUP_LIST, LDAP_GROUPS_FROM_ATTRIBUTES. Simple is a group loaded from LDAP filter which returns subject ids or identifiers. Group list is an LDAP filter which returns group objects, and the group objects have a list of subjects. Groups from attributes is an LDAP filter that returns subjects which have a multi-valued attribute e.g. affiliations where groups will be created based on subject who have each attribute value	grouperLoaderLdap attribute assignment	Enum	LDAP_SIMPLE
grouperLoaderLdapServerId	Grouper loader LDAP server ID	required	Server ID that is configured in the grouper-loader.properties that identifies the connection information to the LDAP server. Note, if you use "dn", and dn is not an attribute of the object, then the fully qualified object name will be used	grouperLoaderLdap attribute assignment	String	personLdap (note: depends on your configuration)

grouperLoad erLdapFilter	Grouper loader LDAP filter	required	LDAP filter returns objects that have subjectIds or subjectIdentifiers and group name (if LDAP_GROUP_LIST). (v2.6.9+) If this starts and ends like a jexl script, it will evaluate as a jexl script, e.g. <pre> \${ var now = (System.currentTimeMillis() + 11644473600000L) * 10000; '(&(accountexpires<=' + now + '))'; } \${ var now = grouperUtil.ldapAdDateFromMillis1970 (java.lang.System.currentTimeMillis()); '(&(accountexpires<=' + now + '))'; } \${ var now = grouperUtil.ldapAdDateCurrent(); '(&(accountexpires<=' + now + '))'; } \${'(&(accountexpires<=' + grouperUtil. ldapAdDateCurrent() + '))'}</pre>	grouperLoaderL dap attribute assignment	String	(affiliation=st udent)
grouperLoad erLdapSubje ctAttribute	Grouper loader LDAP subject attribute name	required, for LDAP_SIMPLE, and LDAP_GROUP_ LIST, optional for LDAP GROU PS_FROM_ATT RIBUTES	Attribute name of the filter object result that holds the subject id.	grouperLoaderL dap attribute assignment	String	hasMember, or personId
grouperLoad erLdapGroup Attribute	Grouper loader LDAP group attribute name	required for LDAP_GROUPS _FROM_ATTRIB UTES	Attribute name of the filter object result that holds the group name. Note, in 2.1.5+ you can put multiple attribute names here comma separated	grouperLoaderL dap attribute assignment	String	affiliation
grouperLoad erLdapGroup AttributeProc essingExpres sion	Grouper loader LDAP group attribute processing expression	optional for LDAP _GROUPS_FRO M_ATTRIBUTES	Expression to process group name to potentially split it into other names or get the name in there	grouperLoaderL dap attribute assignment	String	<pre> \${GrouperUtil.escapeSplitTrimCurlyColons(groupExtension, "dept Description", "[a-zA-Z0-9_]", "-")}</pre>
grouperLoad erLdapSearc hDn	Grouper loader LDAP search base DN	optional	Location that constrains the subtree where the filter is applicable. Note, this is relative to the base DN in the ldap server config in the grouper-loader.properties for this server. This makes the query more efficient	grouperLoaderL dap attribute assignment	String	ou=people

grouperLoad erLdapQuartz Cron	Grouper loader LDAP quartz cron	required	Quartz cron config string, e.g. every day at 8am is: 0 0 8 * * ? Here are more examples	grouperLoaderL dap attribute assignment	String	0 0 8 * * ?
grouperLoad erLdapSourc eId	Grouper loader LDAP source ID	optional	Source ID from the sources.xml that narrows the search for subjects. This is optional though makes the loader job more efficient	grouperLoaderL dap attribute assignment	String	schoolPeople
grouperLoad erLdapSubje ctIdType	Grouper loader LDAP subject ID type	optional	The type of subject ID. This can be either: subjectId (most efficient, default), subjectIdentifier (2nd most efficient), or subjectIdOrIdentifier	grouperLoaderL dap attribute assignment	Enum	subjectId, subjectIdentif ier, subjectIdOrId entifier
grouperLoad erLdapSearc hScope	Grouper loader LDAP search scope	optional	How the deep in the subtree the search will take place. Can be OBJECT_SCOPE, ONELEVEL_SCOPE, or SUBTREE_SCOPE (default)	grouperLoaderL dap attribute assignment	Enum	OBJECT_SC OPE, ONELEVEL_ SCOPE, SUBTREE_S COPE
grouperLoad erLdapAndGr oups	Grouper loader LDAP require in groups	optional	If you want to restrict membership in the dynamic group based on other group(s), put the list of group names here comma-separated. The require groups means if you put a group names in there (e.g. school:community:employee) then it will 'and' that group with the member list from the loader. So only members of the group from the loader query who are also employees will be in the resulting group	grouperLoaderL dap attribute assignment	String	school: community: employee
grouperLoad erLdapPriority	Grouper loader LDAP schedul ing priority	optional	Quartz has a fixed threadpool (max configured in the grouper-loader.properties), and when the max is reached, then jobs are prioritized by this integer. The higher the better, and the default if not set is 5.	grouperLoaderL dap attribute assignment	Integer	5
grouperLoad erLdapGroup sLike	Grouper loader LDAP groups like	optional, for LDAP_GROUP_ LIST, or LDAP_GROUPS _FROM_ATTRIB UTES	This should be a sql like string (e.g. school:orgs:%org%_systemOfRecord), and the loader should be able to query group names to see which names are managed by this loader job. So if a group falls off the loader resultset (or is moved), this will help the loader remove the members from this group. Note, if the group is used anywhere as a member or composite member, it wont be removed. All include/exclude/requireGroups will be removed. Though the two groups, include and exclude, will not be removed if they have members. There is a grouper-loader.properties setting to remove loader groups if empty and not used: # if using a sql table, and specifying the name like string, then should the group (in addition to memberships) # be removed if not used anywhere else? loader.sqlTable.likeString.removeGroupIfNotUsed = true	grouperLoaderL dap attribute assignment	String	school: orgs:%org% _systemOfRe cord
grouperLoad erLdapExtraA ttributes	Grouper loader LDAP extra attributes	optional, for LDAP_GROUP_ LIST	Attribute names (comma separated) to get LDAP data for expressions in group name, displayExtension, description	grouperLoaderL dap attribute assignment	String	name, description
grouperLoad erLdapAttribu teFilterExpre ssion (2.1.4+)	Grouper loader LDAP JEXL expression to filter attributes in LDAP_GROUP S_FROM_ATT RIBUTES	optional	The value is a JEXL expression with the variable: attributeValue e.g. to specify a few values: \${attributeValue == 'a' attributeValue == 'b'} e.g. to restrict a few values: \${attributeValue != 'a' && attributeValue != 'b'} e.g. to use some java methods: \${attributeName.toLowerCase().startsWith('st')} e.g. to do a regex: \${attributeName =~ '^fa.*\$' }	grouperLoaderL dap attribute assignment	String	<code>\${att ribut eVal ue == 'a' attri buteV alue == 'b'}</code>
grouperLoad erLdapResult sTransformationClass (2.4.0 with patches)	Grouper loader LDAP results transformation class	optional for LDAP _GROUPS_FRO M_ATTRIBUTES	Custom Java class to adjust data from LDAP in order to support advanced data transformations and filtering. For example, if your LDAP attribute contains a delimited string and you need to parse it to form multiple groups. Your class should extend LdapResultsTransformationBase. See LdapResultsTransformationDelimitedValueExample as an example.	grouperLoaderL dap attribute assignment	String	edu. internet2. middleware. grouper.app. loader.Ldap.L dapResultsTr ansformation DelimitedVal ueExample
grouperLoad erLdapGroup NameExpres sion	Grouper loader LDAP group name expression	optional, for LDAP_GROUP_ LIST, or LDAP_GROUPS _FROM_ATTRIB UTES	JEXL expression language fragment that evaluates to the group name (relative to the stem of the group which has the loader definition). groupAttributes['dn'] is a variable in scope as is groupAttributes['cn'] etc Note, if you set loader.ldap.requireTopStemAsStemFromConfigGroup=false in the grouper-loader.properties, it will use the name for all ldap loader jobs from the root of the repository, not relative to the loader group. Note: if using LDAP_GROUPS_FROM_ATTRIBUTES then use the variable groupAttribute	grouperLoaderL dap attribute assignment	String	

```
someF
older
:{$gr
oupAt
tribu
tes
['nam
e']}
```

```
group
s:{$l
oader
LdapE
lutil
s.
conve
rtDnT
oSubP
ath
(group
pAttr
ibute
s
['dn'
],
null,
null)}
```

You can
massage the
data with
java regex, e.
g. replace
non
alphanumeric
(or dash)
with
underscore

```
{$gro
upAtt
ribut
es
['nam
e'].
repla
ceAll
("[^a
-zA-
Z0-
9_-
]",
"_")}
```

Or for
LDAP_GRO
UPS_FROM
_ATTRIBUT
ES

						<pre> \${groupAttribute.replaceAll("[^a-zA-Z0-9_-]", "-")} </pre>
grouperLoaderLdapGroupDisplayNameExpression	Grouper loader LDAP group display name expression	optional, for LDAP_GROUP_LIST, or LDAP_GROUPS_FROM_ATTRIBUTES	JEXL expression language fragment that evaluates to the group display name. groupAttributes['dn'] is a variable in scope as is groupAttributes['cn'] etc	grouperLoaderLdap attribute assignment	String	<pre> Some folder: \${groupAttributes['displayname']} </pre>
grouperLoaderLdapGroupDescriptionExpression	Grouper loader LDAP group description expression	optional, for LDAP_GROUP_LIST, or LDAP_GROUPS_FROM_ATTRIBUTES	JEXL expression language fragment that evaluates to the group description. groupAttributes['dn'] is a variable in scope as is groupAttributes['cn'] etc Note, descriptions for groups may default to "<extension> auto-created by grouperLoader" unless you set loader.allowBlankGroupDescriptions to true in grouper-loader.properties (it is false by default)	grouperLoaderLdap attribute assignment	String	<pre> Auto-created based on LDAP group \${dn} </pre>
grouperLoaderLdapSubjectExpression	Grouper loader LDAP subject expression	optional	JEXL expression language fragment that processes the subject string before passing it to the subject API	grouperLoaderLdap attribute assignment	String	<pre> \${loaderLdapUtils.convertDnToSpecificValue(subjectId)} </pre>
grouperLoaderLdapGroupTypes	Grouper loader LDAP group types	optional for LDAP_GROUP_LIST or LDAP_GROUPS_FROM_ATTRIBUTES, not considered for LDAP_SIMPLE	Comma separated GroupTypes which will be applied to the loaded groups. The reason this enhancement exists is so we can do a group list filter and attach addIncludeExclude to the groups. Note, if you do this (or use some requireGroups), the group name in the loader query should end in the system of record suffix, which by default is _systemOfRecord.	grouperLoaderLdap attribute assignment	String	addIncludeExclude
grouperLoaderLdapReaders	Grouper loader LDAP group readers	optional for LDAP_GROUP_LIST or LDAP_GROUPS_FROM_ATTRIBUTES	Comma separated subjectIds or subjectIdentifiers who will be allowed to READ the group memberships.	grouperLoaderLdap attribute assignment	String	school:app:someApp:someAppReaders
grouperLoaderLdapViewers	Grouper loader LDAP group viewers	optional for LDAP_GROUP_LIST or LDAP_GROUPS_FROM_ATTRIBUTES	Comma separated subjectIds or subjectIdentifiers who will be allowed to VIEW the group.	grouperLoaderLdap attribute assignment	String	school:app:someApp:someAppViewers

grouperLoad erLdapAdmins	Grouper loader LDAP group admins	optional for LDAP_GROUP_ LIST or LDAP_GROUPS _FROM_ATTRIB UTES	Comma separated subjectIds or subjectIdentifiers who will be allowed to ADMIN the group (view, read, update, delete, rename, etc).	grouperLoaderL dap attribute assignment	String	school:app: someApp: someAppAd mins
grouperLoad erLdapUpdat ers	Grouper loader LDAP group updaters	optional for LDAP_GROUP_ LIST or LDAP_GROUPS _FROM_ATTRIB UTES	Comma separated subjectIds or subjectIdentifiers who will be allowed to UPDATE the group memberships.	grouperLoaderL dap attribute assignment	String	school:app: someApp: someAppUpd aters
grouperLoad erLdapOptins	Grouper loader LDAP group optins	optional for LDAP_GROUP_ LIST or LDAP_GROUPS _FROM_ATTRIB UTES	Comma separated subjectIds or subjectIdentifiers who will be allowed to OPTIN self membership of the group.	grouperLoaderL dap attribute assignment	String	school:app: someApp: someAppOpti ns
grouperLoad erLdapOptouts	Grouper loader LDAP group optouts	optional for LDAP_GROUP_ LIST or LDAP_GROUPS _FROM_ATTRIB UTES	Comma separated subjectIds or subjectIdentifiers who will be allowed to OPTOUT self membership of the group.	grouperLoaderL dap attribute assignment	String	school:app: someApp: someAppOpt outs

LDAP server configuration

Configure LDAP servers in the grouper-loader.properties:

```
#####
## LDAP connections
#####
# specify the ldap connection with user, pass, url
# the string after "ldap." is the ID of the connection, and it should not have
# spaces or other special chars in it. In this case is it "personLdap"

#note the URL should start with ldap: or ldaps: if it is SSL.
#It should contain the server and port (optional if not default), and baseDn,
#e.g. ldaps://ldapserverschool.edu:636/dc=school,dc=edu
#ldap.personLdap.url = ldaps://ldapserverschool.edu:636/dc=school,dc=edu

#optional, if authenticated
#ldap.personLdap.user = uid=someapp,ou=people,dc=myschool,dc=edu

#optional, if authenticated note the password can be stored encrypted in an external file
#ldap.personLdap.pass = secret

#optional, if you are using tls, set this to true. Generally you will not be using an SSL URL to use TLS...
#ldap.personLdap.tls = false

#optional, if using sasl
#ldap.personLdap.saslAuthorizationId =
#ldap.personLdap.saslRealm =

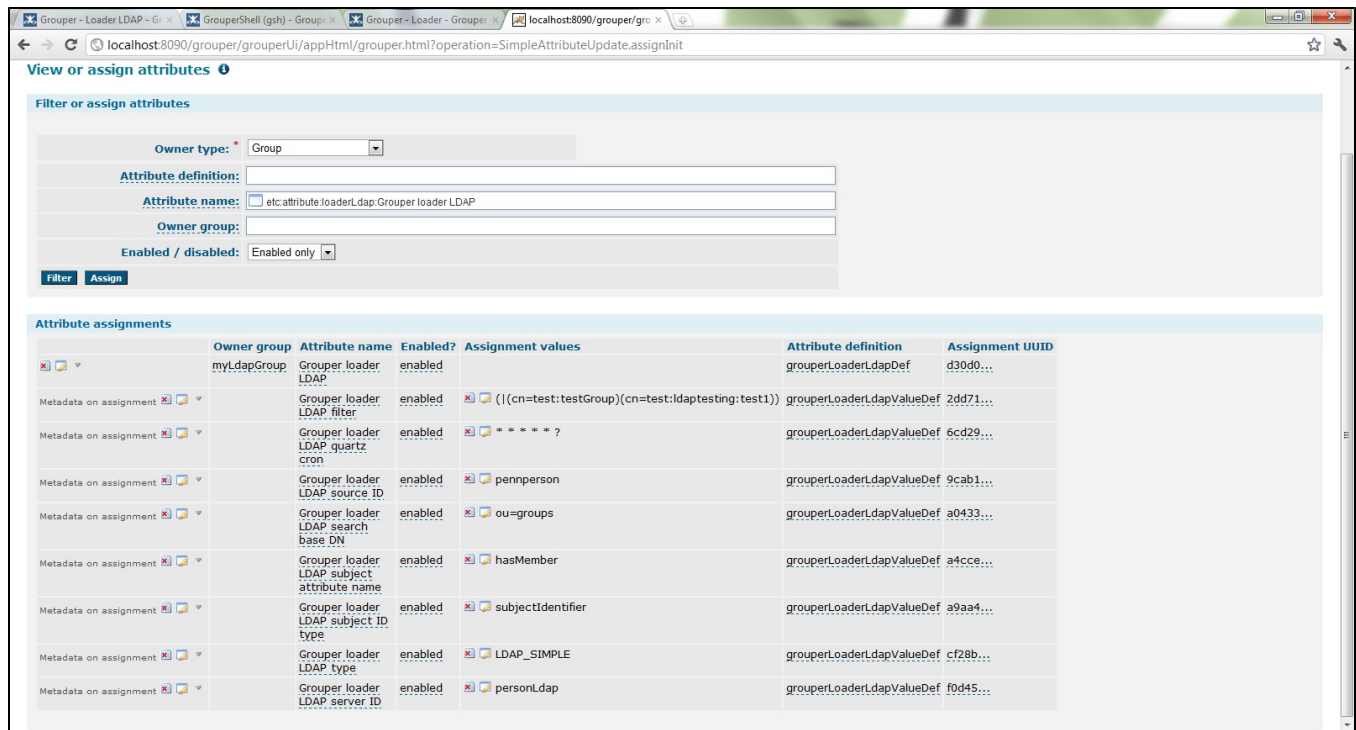
#optional (note, time limit is for search operations, timeout is for connection timeouts),
#most of these default to vt-ldap defaults. times are in millis
#validateOnCheckout defaults to true if all other validate methods are false
#ldap.personLdap.batchSize =
#ldap.personLdap.countLimit =
#ldap.personLdap.timeLimit =
#ldap.personLdap.timeout =
#ldap.personLdap.minPoolSize =
#ldap.personLdap.maxPoolSize =
#ldap.personLdap.validateOnCheckIn =
#ldap.personLdap.validateOnCheckOut =
#ldap.personLdap.validatePeriodically =
#ldap.personLdap.validateTimerPeriod =
#ldap.personLdap.pruneTimerPeriod =
#if connections expire after a certain amount of time, this is it, in millis, defaults to 300000 (5 minutes)
#ldap.personLdap.expirationTime =
```

Here is an example config], with serverId: myLdap

```
ldap.myLdap.user = uid=someUser,ou=people,dc=school,dc=edu
ldap.myLdap.pass = abcdabcd
ldap.myLdap.url = ldaps://ldap.school.edu:636/dc=school,dc=edu
```

Configuration

You can configure the loader LDAP attributes with the lite UI, GSH (example below), or WS. Here is a screenshot of the UI



Note that as of Grouper 2.3, if you create a new loader job, you can have it scheduled without restarting the daemon by using an option in the New UI. If you're an admin of the group, under "More actions", there's an option named "Schedule loader process". This option can also be used if you change the loader's schedule.

Configure LDAP loader settings in grouper-loader.properties

```
#####
## LDAP loader settings
#####

# el classes to add to the el context for the EL to calculate subejct ids or group names etc.
# Comma-separated fully qualified classnamesm will be registered by the non-fully qualified
# uncapitalized classname. So you register a.b.SomeClass, it will be available by variable: someClass
loader.ldap.el.classes =
```

Loader LDAP expression language

The JEXL context and examples are explained here

The expressions expected in the attributes: grouperLoaderLdapGroupNameExpression, grouperLoaderLdapGroupDisplayExtensionExpression, grouperLoaderLdapGroupDescriptionExpression, grouperLoaderLdapSubjectExpression.

The expressions do not need to be surrounded by

```
${}
```

since you might be appending strings

You can register your own variables so your own logic can be run to process the information with the grouper-loader.properties setting: loader.ldap.el.classes

Variable	Represents	When set
subjectAttributes ['subjectId']	The subject id, identifier, or idOrIdentifier	When processing the subject. e.g. if you have a subjectAttribute config, it will be here
loaderLdapElUtils	The LoaderLdapElUtils class	Always

Here are examples:

Convert a DN to the most specific value, e.g. from cn=something,ou=groups,dn=school,dn=edu, to: something In this case, it converts the subjectId to something the subject API can use

```
${loaderLdapElUtils.convertDnToSpecificValue(subjectId)}
```

sdf

Test case common setup

Find two groups in your LDAP that do not have many members

In this case, here is a group in ldap:

```
cn=test:ldapTesting:test1,ou=groups,dc=upenn,dc=edu
```

With hasMember attribute values:

```
netmon
```

There is another group in ldap:

```
cn=test:testGroup,ou=groups,dc=upenn,dc=edu
```

With hasMember attribute values:

```
choate
mchyzer
harveycg
mchyzer
```

There is another group in ldap:

```
cn=test:testGroup2,ou=groups,dc=upenn,dc=edu
```

With hasMember attribute values:

```
taberkow
choate
bwh
convery
```

Make sure the member ids in LDAP are subject ids or identifiers in Grouper. In this case, I am using a new instance of Grouper, not my institutional one, so I will just make a new source, and insert dummy data, with subject identifiers (even though subject id's are more efficient)

Run this SQL (note, this is tested in mysql, but will work with small adjustment in any db)

```
CREATE TABLE `person_source_v` (
  `penn_id` VARCHAR(50) NOT NULL,
  `name` VARCHAR(50) DEFAULT NULL,
  `email` VARCHAR(200) DEFAULT NULL,
  `description` VARCHAR(50) DEFAULT NULL,
  `pennname` VARCHAR(50) DEFAULT NULL,
  `description_lower` VARCHAR(50) DEFAULT NULL,
  PRIMARY KEY (`penn_id`)
);
INSERT INTO `person_source_v`(`penn_id`,`name`,`description`,`pennname`,`description_lower`) VALUES
('44567890','Chris Hyzer','Chris Hyzer','mchyzer','chris hyzer, mchyzer, 44567890'),
('11234567','John Smith','John Smith','jsmith','john smith, jsmith, 11234567'),
('12345678','Bryan Hall','Bryan Hall','bwh','bryan hall, bwh, 12345678'),
('10000000','netmon','netmon','netmon','netmon, 10000000'),
('22345678','Bill Choate','Bill Choate','choate','bill choate, choate, 22345678'),
('33456789','Craig Harvey','Craig Harvey','harveycg','craig harvey, harveycg, 33456789')
;

COMMIT;
```

Here is the sources.xml config:

```
<source adapterClass="edu.internet2.middleware.subject.provider.JDBCSourceAdapter2">

  <id>pennperson</id>
  <name>Penn person</name>
  <type>person</type>
  <init-param>
    <param-name>jdbcConnectionProvider</param-name>
    <param-value>edu.internet2.middleware.grouper.subj.GrouperJdbcConnectionProvider</param-value>
  </init-param>
  <init-param>
    <param-name>maxResults</param-name>
    <param-value>1000</param-value>
  </init-param>

  <init-param>
    <param-name>dbTableOrView</param-name>
    <param-value>person_source_v</param-value>
  </init-param>
  <init-param>
    <param-name>subjectIdCol</param-name>
    <param-value>penn_id</param-value>
  </init-param>
  <init-param>
    <param-name>nameCol</param-name>
    <param-value>name</param-value>
  </init-param>
  <init-param>
    <param-name>descriptionCol</param-name>
    <param-value>description</param-value>
  </init-param>
  <init-param>
    <!-- search col where general searches take place, lower case -->
    <param-name>lowerSearchCol</param-name>
    <param-value>description_lower</param-value>
  </init-param>
  <init-param>
    <!-- optional col if you want the search results sorted in the API (note, UI might override) -->
    <param-name>defaultSortCol</param-name>
    <param-value>description</param-value>
  </init-param>
  <init-param>
    <!-- col which identifies the row, perhaps not subjectId -->
    <param-name>subjectIdentifierCol0</param-name>
```

```

    <param-value>pennname</param-value>
</init-param>
<init-param>
  <param-name>subjectIdentifierCol1</param-name>
  <param-value>penn_id</param-value>
</init-param>
<!-- now you can count up from 0 to N of attributes for various cols -->
<init-param>
  <param-name>subjectAttributeCol0</param-name>
  <param-value>pennname</param-value>
</init-param>
<init-param>
  <param-name>subjectAttributeName0</param-name>
  <param-value>PENNNNAME</param-value>
</init-param>
<init-param>
  <param-name>subjectAttributeCol1</param-name>
  <param-value>email</param-value>
</init-param>
<init-param>
  <param-name>subjectAttributeName1</param-name>
  <param-value>EMAIL</param-value>
</init-param>
<init-param>
  <param-name>sortAttribute0</param-name>
  <param-value>description</param-value>
</init-param>
<init-param>
  <param-name>searchAttribute0</param-name>
  <param-value>description_lower</param-value>
</init-param>
</source>

```

Configure an LDAP connection in the grouper-loader.properties:

```

ldap.personLdap.url = ldaps://penngroups.upenn.edu/dc=upenn,dc=edu
ldap.personLdap.user = uid=someone,ou=entities,dc=upenn,dc=edu
ldap.personLdap.pass = xxxxxxxx

```

LDAP_SIMPLE test case

First do the common setup above, then you can configure and run the loader in GSH:

```

gsh 0% grouperSession = GrouperSession.startRootSession();
gsh 1% group = new GroupSave(grouperSession).assignName("someStem:myLdapGroup").
assignCreateParentStemsIfNotExist(true).save();
gsh 2% attributeAssign = group.getAttributeDelegate().assignAttribute(LoaderLdapUtils.
grouperLoaderLdapAttributeDefName()).getAttributeAssign();
#in case you need to retrieve again, here is an example
gsh 2% attributeAssign = group.getAttributeDelegate().retrieveAssignment(null, LoaderLdapUtils.
grouperLoaderLdapAttributeDefName(), false, true);
gsh 3% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapTypeName(),
"LDAP_SIMPLE");
gsh 4% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapFilterName(),
"(|(cn=test:testGroup)(cn=test:ldaptesting:test1))");
# every minute so it is easy to test
gsh 5% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapQuartzCronName(), "0 * * * * ?");
gsh 6% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSearchDnName(),
"ou=groups");
gsh 7% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapServerIdName(),
"personLdap");
gsh 8% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSourceIdName(),
"pennperson");
gsh 10% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectAttributeName(), "hasMember");
# NOTE: hopefully you can use subjectId instead, it will improve the performance a LOT!
gsh 11% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectIdTypeName(), "subjectIdentifier");
gsh 12% group = GroupFinder.findByName(grouperSession, "someStem:myLdapGroup");
gsh 13% loaderRunOneJob(group);
loader ran successfully, inserted 5 memberships, deleted 0 memberships, total membership count: 5
gsh 14% getMembers("someStem:myLdapGroup");
member: id='22345678' type='person' source='pennperson' uuid='360802a1bdf341859109c086ffe79022'
member: id='33456789' type='person' source='pennperson' uuid='5dd1fc0431214a6fa53bf3cb7790d5ea'
member: id='44567890' type='person' source='pennperson' uuid='8b26f3fb43da4661946282227580d5be'
member: id='12345678' type='person' source='pennperson' uuid='db43860f64004ec295129cde994a450d'
member: id='10000000' type='person' source='pennperson' uuid='ea9b420cca1f43b1a1cb8b682cb3624a'
gsh 5% delMember("someStem:myLdapGroup", "22345678");
true
gsh 16% addMember("someStem:myLdapGroup", "GrouperSystem");
true
gsh 17% loaderRunOneJob(group);

loader ran successfully, inserted 1 memberships, deleted 1 memberships, total membership count: 5
gsh 18%

```

Script ready to be copy/pasted to gsh:

```

grouperSession = GrouperSession.startRootSession();
group = new GroupSave(grouperSession).assignName("someStem:myLdapGroup").assignCreateParentStemsIfNotExist(
true).save();
attributeAssign = group.getAttributeDelegate().assignAttribute(LoaderLdapUtils.
grouperLoaderLdapAttributeDefName()).getAttributeAssign();
attributeAssign = group.getAttributeDelegate().retrieveAssignment(null, LoaderLdapUtils.
grouperLoaderLdapAttributeDefName(), false, true);
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapTypeName(),
"LDAP_SIMPLE");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapFilterName(), "(|
(cn=test:testGroup)(cn=test:ldaptesting:test1))");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapQuartzCronName(), "0 *
* * * ?");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSearchDnName(),
"ou=groups");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapServerIdName(),
"personLdap");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSourceIdName(),
"pennperson");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSubjectAttributeName(),
"hasMember");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSubjectIdTypeName(),
"subjectIdentifier");
group = GroupFinder.findByName(grouperSession, "someStem:myLdapGroup");
loaderRunOneJob(group);
getMembers("someStem:myLdapGroup");
delMember("someStem:myLdapGroup", "22345678");
addMember("someStem:myLdapGroup", "GrouperSystem");
loaderRunOneJob(group);

```

LDAP_GROUP_LIST test case

First do the common setup above, then you can configure and run the loader in GSH:

```

gsh 0% grouperSession = GrouperSession.startRootSession();
gsh 1% group = new GroupSave(grouperSession).assignName("anotherStem:groupListLdapGroup").
assignCreateParentStemsIfNotExist(true).save();
gsh 2% attributeAssign = group.getAttributeDelegate().assignAttribute(LoaderLdapUtils.
grouperLoaderLdapAttributeDefName()).getAttributeAssign();
#in case you need to retrieve again, here is an example
gsh 2% attributeAssign = group.getAttributeDelegate().retrieveAssignment(null, LoaderLdapUtils.
grouperLoaderLdapAttributeDefName(), false, true);
gsh 3% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapTypeName(),
"LDAP_GROUP_LIST");
gsh 4% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapFilterName(),
"(|(cn=test:testGroup)(cn=test:ldapTesting:test1)(cn=test:testEmptyGroup))");
# every minute so it is easy to test
gsh 5% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapQuartzCronName(), "0 * * * * ?");
gsh 6% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSearchDnName(),
"ou=groups");
gsh 7% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapServerIdName(),
"personLdap");
gsh 8% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSourceIdName(),
"pennperson");
gsh 10% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectAttributeName(), "hasMember");
# NOTE: hopefully you can use subjectId instead, it will improve the performance a LOT!
gsh 11% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectIdTypeName(), "subjectIdentifier");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapExtraAttributesName(), "cn");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapGroupNameExpressionName(), "groups:${groupAttributes['cn']}");
gsh 13% group = GroupFinder.findByName(grouperSession, "anotherStem:groupListLdapGroup");
gsh 13% loaderRunOneJob(group);
loader ran successfully, inserted 5 memberships, deleted 0 memberships, total membership count: 5
gsh 14% getGroups("anotherStem")
group: name='anotherStem:groups:test:testGroup' displayName='anotherStem:groups:test:testGroup'
uuid='84cf356b1f694594971439ade9c32ce8'
group: name='anotherStem:groupListLdapGroup' displayName='anotherStem:groupListLdapGroup'
uuid='e42924f906ff4c0cb97bc8766338fea4'
group: name='anotherStem:groups:test:ldapTesting:test1' displayName='anotherStem:groups:test:ldapTesting:test1'
uuid='e767dcdf3d7340c38986ed4cdab30c44'
gsh 15% getMembers("anotherStem:groups:test:testGroup");
member: id='22345678' type='person' source='pennperson' uuid='360802a1bdf341859109c086ffe79022'
member: id='33456789' type='person' source='pennperson' uuid='5dd1fc0431214a6fa53bf3cb7790d5ea'
member: id='44567890' type='person' source='pennperson' uuid='8b26f3fb43da4661946282227580d5be'
member: id='12345678' type='person' source='pennperson' uuid='db43860f64004ec295129cde994a450d'
gsh 15% getMembers("anotherStem:groups:test:ldapTesting:test1");
member: id='10000000' type='person' source='pennperson' uuid='ea9b420cca1f43b1a1cb8b682cb3624a'
gsh 15% delMember("anotherStem:groups:test:testGroup", "22345678");
true
gsh 16% addMember("anotherStem:groups:test:ldapTesting:test1", "GrouperSystem");
true
gsh 17% loaderRunOneJob(group);

loader ran successfully, inserted 1 memberships, deleted 1 memberships, total membership count: 5
gsh 18%

```

Script ready to copy pasted into GSH:

```

grouperSession = GrouperSession.startRootSession();
group = new GroupSave(grouperSession).assignName("anotherStem:groupListLdapGroup").
assignCreateParentStemsIfNotExist(true).save();
attributeAssign = group.getAttributeDelegate().assignAttribute(LoaderLdapUtils.
grouperLoaderLdapAttributeDefName()).getAttributeAssign();
attributeAssign = group.getAttributeDelegate().retrieveAssignment(null, LoaderLdapUtils.
grouperLoaderLdapAttributeDefName(), false, true);
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapTypeName(),
"LDAP_GROUP_LIST");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapFilterName(), "(|
(cn=test:testGroup)(cn=test:ldaptesting:test1)(cn=test:testEmptyGroup)");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapQuartzCronName(), "0 *
* * * ?");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSearchDnName(),
"ou=groups");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapServerIdName(),
"personLdap");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSourceIdName(),
"pennperson");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSubjectAttributeName(),
"hasMember");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSubjectIdTypeName(),
"subjectIdentifier");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapExtraAttributesName(),
"cn");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapGroupNameExpressionName(), "groups:${groupAttributes['cn']}");
group = GroupFinder.findByName(grouperSession, "anotherStem:groupListLdapGroup");
loaderRunOneJob(group);
getGroups("anotherStem")
getMembers("anotherStem:groups:test:testGroup");
getMembers("anotherStem:groups:test:ldaptesting:test1");
delMember("anotherStem:groups:test:testGroup", "22345678");
addMember("anotherStem:groups:test:ldaptesting:test1", "GrouperSystem");
loaderRunOneJob(group);

```

These lines will return either the subjectId or a groupId which it will create if not there already

```

attributeAssign.getAttributeValueDelegate().deleteValue(LoaderLdapUtils.grouperLoaderLdapSourceIdName(),
"pennperson");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectExpressionName(), "${loaderLdapElUtils.convertDnToSpecificValueTest(subjectId,
'convery', 'test:convery')}");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSubjectIdTypeName(),
"subjectId");

```

LDAP_GROUPS_FROM_ATTRIBUTES test case

I dont have a multi-valued user attribute, I will just do an inverted loader job from the above... the group ldap objects are the subjects (by identifier), and the hasmember multi-valued attributes will be the groups :)

```

gsh 0% grouperSession = GrouperSession.startRootSession();
gsh 1% group = new GroupSave(grouperSession).assignName("yetAnotherStem:groupsFromAttributesLdapGroup").
assignCreateParentStemsIfNotExist(true).save();
gsh 2% new GroupSave(grouperSession).assignName("test:testGroup").assignCreateParentStemsIfNotExist(true).
save();
gsh 3% new GroupSave(grouperSession).assignName("test:testGroup2").assignCreateParentStemsIfNotExist(true).
save();

gsh 2% attributeAssign = group.getAttributeDelegate().assignAttribute(LoaderLdapUtils.
grouperLoaderLdapAttributeDefName()).getAttributeAssign();
#in case you need to retrieve again, here is an example
gsh 2% attributeAssign = group.getAttributeDelegate().retrieveAssignment(null, LoaderLdapUtils.
grouperLoaderLdapAttributeDefName(), false, true);
gsh 3% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapTypeName(),
"LDAP_GROUPS_FROM_ATTRIBUTES");

```

```

gsh 4% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapFilterName(),
"(|(cn=test:testGroup)(cn=test:testGroup2))");
# every minute so it is easy to test
gsh 5% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapQuartzCronName(), "0 * * * * ?");
gsh 6% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSearchDnName(),
"ou=groups");
gsh 7% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapServerIdName(),
"personLdap");
gsh 8% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSourceIdName(),
"g:gsa");
gsh 8% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapGroupAttributeName(), "hasmember");

# NOTE: hopefully you can use subjectId instead, it will improve the performance a LOT!
gsh 11% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectIdTypeName(), "subjectIdentifier");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapGroupNameExpressionName(), "groups:${groupAttribute}");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectExpressionName(), "${subjectAttributes['cn']}");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapExtraAttributesName(), "cn");
gsh 13% group = GroupFinder.findByName(grouperSession, "yetAnotherStem:groupsFromAttributesLdapGroup");
gsh 13% loaderRunOneJob(group);
loader ran successfully, inserted 8 memberships, deleted 0 memberships, total membership count: 8
gsh 14% getGroups("yetAnotherStem:groups")
group: name='yetAnotherStem:groups:convery' displayName='yetAnotherStem:groups:convery'
uuid='0a8c93cc17914c44b8ae24e7560c4833'
group: name='yetAnotherStem:groups:harveycg' displayName='yetAnotherStem:groups:harveycg'
uuid='2688ff9831bc4c15b8c49095a85d432e'
group: name='yetAnotherStem:groups:choate' displayName='yetAnotherStem:groups:choate'
uuid='3381dd9d59c041289596ae7b9f4ea38e'
group: name='yetAnotherStem:groups:taberkow' displayName='yetAnotherStem:groups:taberkow'
uuid='343af34411144fbfbdca0482edec9435'
group: name='yetAnotherStem:groups:mchyzer' displayName='yetAnotherStem:groups:mchyzer'
uuid='631bf71674b146c98abefcf605cc710b'
group: name='yetAnotherStem:groupsFromAttributesLdapGroup' displayName='yetAnotherStem:
groupsFromAttributesLdapGroup' uuid='9dd5e5eca4f04400965ed3bc0b986ec2'
group: name='yetAnotherStem:groups:bwh' displayName='yetAnotherStem:groups:bwh'
uuid='efaa40a58bae4cc7b04380af11db78f6'
gsh 15% getMembers("yetAnotherStem:groups:choate");
member: id='22345678' type='person' source='pennperson' uuid='360802a1bdf341859109c086ffe79022'
member: id='33456789' type='person' source='pennperson' uuid='5dd1fc0431214a6fa53bf3cb7790d5ea'
member: id='44567890' type='person' source='pennperson' uuid='8b26f3fb43da4661946282227580d5be'
member: id='12345678' type='person' source='pennperson' uuid='db43860f64004ec295129cde994a450d'
gsh 15% for (theGroup : getMembers("yetAnotherStem:groups:choate")) {print(theGroup.getName());}
test:testGroup
test:testGroup2
gsh 15% delMember("yetAnotherStem:groups:choate", "test:testGroup");
true
gsh 16% addMember("yetAnotherStem:groups:choate", "GrouperSystem");
true
gsh 17% loaderRunOneJob(group);

loader ran successfully, inserted 1 memberships, deleted 1 memberships, total membership count: 8
gsh 18%

```

Script which can be run from GSH:

```

grouperSession = GrouperSession.startRootSession();
group = new GroupSave(grouperSession).assignName("yetAnotherStem:groupsFromAttributesLdapGroup").
assignCreateParentStemsIfNotExist(true).save();
new GroupSave(grouperSession).assignName("test:testGroup").assignCreateParentStemsIfNotExist(true).save();
new GroupSave(grouperSession).assignName("test:testGroup2").assignCreateParentStemsIfNotExist(true).save();
attributeAssign = group.getAttributeDelegate().assignAttribute(LoaderLdapUtils.
grouperLoaderLdapAttributeDefName()).getAttributeAssign();
attributeAssign = group.getAttributeDelegate().retrieveAssignment(null, LoaderLdapUtils.
grouperLoaderLdapAttributeDefName(), false, true);
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapTypeName(),
"LDAP_GROUPS_FROM_ATTRIBUTES");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapFilterName(), "(|
(cn=test:testGroup)(cn=test:testGroup2))");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapQuartzCronName(), "0 *
* * * ?");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSearchDnName(),
"ou=groups");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapServerIdName(),
"personLdap");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSourceIdName(), "g:
gsa");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapGroupAttributeName(),
"hasmember");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSubjectIdTypeName(),
"subjectIdentifier");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapGroupNameExpressionName(), "groups:${groupAttribute}");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectExpressionName(), "${subjectAttributes['cn']}");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapExtraAttributesName(),
"cn");
group = GroupFinder.findByName(grouperSession, "yetAnotherStem:groupsFromAttributesLdapGroup");
loaderRunOneJob(group);
getGroups("yetAnotherStem:groups")
getMembers("yetAnotherStem:groups:choate");
for (theGroup : getMembers("yetAnotherStem:groups:choate")) {print(theGroup.getName());}
delMember("yetAnotherStem:groups:choate", "test:testGroup");
addMember("yetAnotherStem:groups:choate", "GrouperSystem");
loaderRunOneJob(group);

```

LDAP_GROUPS_FROM_ATTRIBUTES include/exclude example

I dont have a multi-valued user attribute, I will just do an inverted loader job from the above... the group ldap objects are the subjects (by identifier), and the hasmember multi-valued attributes will be the groups :)

First I make sure these are set in the grouper.properties:

```

#if the addIncludeExclude and requireInGroups should be enabled, and if the type(s) should be
#auto-created, and used to auto create groups to facilitate include and exclude lists, and require lists
grouperIncludeExclude.use = true
grouperIncludeExclude.requireGroups.use = true

```

Setup and run the loader. Note the group name ends in _systemOfRecord

```

grouperSession = GrouperSession.startRootSession();
group = new GroupSave(grouperSession).assignName("yetAnotherStem:groupsFromAttributesLdapGroupIncludeExclude").
assignCreateParentStemsIfNotExist(true).save();
new GroupSave(grouperSession).assignName("test:testGroup").assignCreateParentStemsIfNotExist(true).save();
new GroupSave(grouperSession).assignName("test:testGroup2").assignCreateParentStemsIfNotExist(true).save();
attributeAssign = group.getAttributeDelegate().assignAttribute(LoaderLdapUtils.
grouperLoaderLdapAttributeDefName()).getAttributeAssign();
attributeAssign = group.getAttributeDelegate().retrieveAssignment(null, LoaderLdapUtils.
grouperLoaderLdapAttributeDefName(), false, true);
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapTypeName(),
"LDAP_GROUPS_FROM_ATTRIBUTES");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapFilterName(), "(|
(cn=test:testGroup)(cn=test:testGroup2))");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapQuartzCronName(), "0 *
* * * ?");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSearchDnName(),
"ou=groups");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapServerIdName(),
"personLdap");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSourceIdName(), "g:
gsa");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapGroupAttributeName(),
"hasmember");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSubjectIdTypeName(),
"subjectIdentifier");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapGroupNameExpressionName(), "groupsIncludeExclude:${groupAttribute}_systemOfRecord");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectExpressionName(), "${subjectAttributes['cn']}");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapExtraAttributesName(),
"cn");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapGroupTypesName(),
"addIncludeExclude");
group = GroupFinder.findByName(grouperSession, "yetAnotherStem:groupsFromAttributesLdapGroupIncludeExclude");
loaderRunOneJob(group);
getGroups("yetAnotherStem:groups")
getMembers("yetAnotherStem:groups:choate");
for (theGroup : getMembers("yetAnotherStem:groups:choate")) {print(theGroup.getName());}
delMember("yetAnotherStem:groups:choate", "test:testGroup");
addMember("yetAnotherStem:groups:choate", "GrouperSystem");
loaderRunOneJob(group);

```

LDAP_GROUPS_FROM_ATTRIBUTES subject attribute test case

Same as LDAP_GROUPS_FROM_ATTRIBUTES above, though there is a subjectAttribute specified, then used in the subject expression

```

gsh 0% grouperSession = GrouperSession.startRootSession();
gsh 1% group = new GroupSave(grouperSession).assignName("yetAnotherStem2:groupsFromAttributesLdapGroup2").
assignCreateParentStemsIfNotExist(true).save();
gsh 2% new GroupSave(grouperSession).assignName("test:testGroup").assignCreateParentStemsIfNotExist(true).
save();
gsh 2% new GroupSave(grouperSession).assignName("test:testGroup2").assignCreateParentStemsIfNotExist(true).
save();

gsh 2% attributeAssign = group.getAttributeDelegate().assignAttribute(LoaderLdapUtils.
grouperLoaderLdapAttributeDefName()).getAttributeAssign();
#in case you need to retrieve again, here is an example
gsh 2% attributeAssign = group.getAttributeDelegate().retrieveAssignment(null, LoaderLdapUtils.
grouperLoaderLdapAttributeDefName(), false, true);
gsh 3% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapTypeName(),
"LDAP_GROUPS_FROM_ATTRIBUTES");
gsh 4% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapFilterName(),
"(|(cn=test:testGroup)(cn=test:testGroup2))");
# every minute so it is easy to test
gsh 5% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapQuartzCronName(), "0 * * * * ?");
gsh 6% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSearchDnName(),

```

```

"ou=groups");
gsh 7% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapServerIdName(),
"personLdap");
gsh 8% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSourceIdName(),
"g:gsa");
gsh 8% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapGroupAttributeName(), "hasmember");

# NOTE: hopefully you can use subjectId instead, it will improve the performance a LOT!
gsh 11% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectIdTypeName(), "subjectIdentifier");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapGroupNameExpressionName(), "groups:${groupAttribute}");

# NOTE: in this case it this is redundant, you could leave it out and have the same result, but if you need to
process, here it is...
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectExpressionName(), "${subjectAttributes['subjectId']}");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectAttributeName(), "cn");
gsh 13% group = GroupFinder.findByName(grouperSession, "yetAnotherStem2:groupsFromAttributesLdapGroup2");
gsh 13% loaderRunOneJob(group);
loader ran successfully, inserted 8 memberships, deleted 0 memberships, total membership count: 8
gsh 14% getGroups("yetAnotherStem2:groups")
group: name='yetAnotherStem2:groups:convery' displayName='yetAnotherStem2:groups:convery'
uuid='0a8c93cc17914c44b8ae24e7560c4833'
group: name='yetAnotherStem2:groups:harveycg' displayName='yetAnotherStem2:groups:harveycg'
uuid='2688ff9831bc4c15b8c49095a85d432e'
group: name='yetAnotherStem2:groups:choate' displayName='yetAnotherStem2:groups:choate'
uuid='3381dd9d59c041289596ae7b9f4ea38e'
group: name='yetAnotherStem2:groups:taberkow' displayName='yetAnotherStem2:groups:taberkow'
uuid='343af34411144fbfbdc0482edec9435'
group: name='yetAnotherStem2:groups:mchyzer' displayName='yetAnotherStem2:groups:mchyzer'
uuid='631bf71674b146c98abefcf605cc710b'
group: name='yetAnotherStem2:groupsFromAttributesLdapGroup' displayName='yetAnotherStem2:
groupsFromAttributesLdapGroup' uuid='9dd5e5eca4f04400965ed3bc0b986ec2'
group: name='yetAnotherStem2:groups:bwh' displayName='yetAnotherStem2:groups:bwh'
uuid='efaa40a58bae4cc7b04380af11db78f6'
gsh 15% getMembers("yetAnotherStem2:groups:choate");
member: id='22345678' type='person' source='pennperson' uuid='360802a1bdf341859109c086ffe79022'
member: id='33456789' type='person' source='pennperson' uuid='5dd1fc0431214a6fa53bf3cb7790d5ea'
member: id='44567890' type='person' source='pennperson' uuid='8b26f3fb43da4661946282227580d5be'
member: id='12345678' type='person' source='pennperson' uuid='db43860f64004ec295129cde994a450d'
gsh 15% for (theGroup : getMembers("yetAnotherStem2:groups:choate")) {print(theGroup.getName());}
test:testGroup
test:testGroup2
gsh 15% delMember("yetAnotherStem2:groups:choate", "test:testGroup");
true
gsh 16% addMember("yetAnotherStem2:groups:choate", "GrouperSystem");
true
gsh 17% loaderRunOneJob(group);

loader ran successfully, inserted 1 memberships, deleted 1 memberships, total membership count: 8
gsh 18%

```

Script ready to be used in GSH:

```

grouperSession = GrouperSession.startRootSession();
group = new GroupSave(grouperSession).assignName("yetAnotherStem:groupsFromAttributesLdapGroup").
assignCreateParentStemsIfNotExist(true).save();
new GroupSave(grouperSession).assignName("test:testGroup").assignCreateParentStemsIfNotExist(true).save();
new GroupSave(grouperSession).assignName("test:testGroup2").assignCreateParentStemsIfNotExist(true).save();

attributeAssign = group.getAttributeDelegate().assignAttribute(LoaderLdapUtils.
grouperLoaderLdapAttributeDefName()).getAttributeAssign();
attributeAssign = group.getAttributeDelegate().retrieveAssignment(null, LoaderLdapUtils.
grouperLoaderLdapAttributeDefName(), false, true);
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapTypeName(),
"LDAP_GROUPS_FROM_ATTRIBUTES");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapFilterName(), "(|
(cn=test:testGroup)(cn=test:testGroup2))");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapQuartzCronName(), "0 *
* * * ?");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSearchDnName(),
"ou=groups");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapServerIdName(),
"personLdap");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSourceIdName(), "g:
gsa");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapGroupAttributeName(),
"hasmember");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSubjectIdTypeName(),
"subjectIdentifier");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapGroupNameExpressionName(), "groups:${groupAttribute}");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectExpressionName(), "${subjectAttributes['cn']}");
attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapExtraAttributesName(),
"cn");
group = GroupFinder.findByName(grouperSession, "yetAnotherStem:groupsFromAttributesLdapGroup");
loaderRunOneJob(group);
getGroups("yetAnotherStem:groups")
getMembers("yetAnotherStem:groups:choate");
for (theGroup : getMembers("yetAnotherStem:groups:choate")) {print(theGroup.getName());}
delMember("yetAnotherStem:groups:choate", "test:testGroup");
addMember("yetAnotherStem:groups:choate", "GrouperSystem");
loaderRunOneJob(group);

```

LDAP_SIMPLE search scope test case

If there is no search DN, and the search scope is one level, then it will not find anyone. Change it to sublevels, and it should find.

```

gsh 0% grouperSession = GrouperSession.startRootSession();
gsh 1% group = new GroupSave(grouperSession).assignName("someStem2:myLdapGroup2").
assignCreateParentStemsIfNotExist(true).save();
gsh 2% attributeAssign = group.getAttributeDelegate().assignAttribute(LoaderLdapUtils.
grouperLoaderLdapAttributeDefName()).getAttributeAssign();
#in case you need to retrieve again, here is an example
gsh 2% attributeAssign = group.getAttributeDelegate().retrieveAssignment(null, LoaderLdapUtils.
grouperLoaderLdapAttributeDefName(), false, true);
gsh 3% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapTypeName(),
"LDAP_SIMPLE");
gsh 4% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapFilterName(),
"(|(cn=test:testGroup)(cn=test:ldaptesting:test1))");
# every minute so it is easy to test
gsh 5% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapQuartzCronName(), "0 * * * * ?");
gsh 7% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapServerIdName(),
"personLdap");
gsh 8% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSourceIdName(),
"pennperson");
gsh 9% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSearchScopeName(), "ONELEVEL_SCOPE");
gsh 10% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectAttributeName(), "hasMember");
# NOTE: hopefully you can use subjectId instead, it will improve the performance a LOT!
gsh 11% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectIdTypeName(), "subjectIdentifier");
gsh 12% group = GroupFinder.findByName(grouperSession, "someStem2:myLdapGroup2");

#### NOTE: since there is one level scope, and no search dn, then no objects will be found...

gsh 13% loaderRunOneJob(group);
loader ran successfully, inserted 0 memberships, deleted 0 memberships, total membership count: 0
gsh 13% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSearchScopeName(), "SUBTREE_SCOPE");
gsh 13% loaderRunOneJob(group);
loader ran successfully, inserted 5 memberships, deleted 0 memberships, total membership count: 5
gsh 14% getMembers("someStem2:myLdapGroup2");
member: id='22345678' type='person' source='pennperson' uuid='360802a1bdf341859109c086ffe79022'
member: id='33456789' type='person' source='pennperson' uuid='5dd1fc0431214a6fa53bf3cb7790d5ea'
member: id='44567890' type='person' source='pennperson' uuid='8b26f3fb43da4661946282227580d5be'
member: id='12345678' type='person' source='pennperson' uuid='db43860f64004ec295129cde994a450d'
member: id='10000000' type='person' source='pennperson' uuid='ea9b420cca1f43b1a1cb8b682cb3624a'
gsh 5% delMember("someStem2:myLdapGroup2", "22345678");
true
gsh 16% addMember("someStem2:myLdapGroup2", "GrouperSystem");
true
gsh 17% loaderRunOneJob(group);

loader ran successfully, inserted 1 memberships, deleted 1 memberships, total membership count: 5
gsh 18%

```

LDAP_SIMPLE and Groups test case

Make sure the subjects are in a group before adding them to the loaded group

```

gsh 0% grouperSession = GrouperSession.startRootSession();
gsh 1% group = new GroupSave(grouperSession).assignName("someStem3:myLdapGroup3").
assignCreateParentStemsIfNotExist(true).save();
gsh 1% employeeGroup = new GroupSave(grouperSession).assignName("school:community:employee").
assignCreateParentStemsIfNotExist(true).save();
gsh 1% addMember("school:community:employee", "22345678");
gsh 1% addMember("school:community:employee", "33456789");
gsh 1% addMember("school:community:employee", "44567890");
gsh 2% attributeAssign = group.getAttributeDelegate().assignAttribute(LoaderLdapUtils.
grouperLoaderLdapAttributeDefName()).getAttributeAssign();
#in case you need to retrieve again, here is an example
gsh 2% attributeAssign = group.getAttributeDelegate().retrieveAssignment(null, LoaderLdapUtils.
grouperLoaderLdapAttributeDefName(), false, true);
gsh 3% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapTypeName(),
"LDAP_SIMPLE");
gsh 4% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapFilterName(),
"(|(cn=test:testGroup)(cn=test:ldaptesting:test1))");
# every minute so it is easy to test
gsh 5% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapQuartzCronName(), "0 * * * * ?");
gsh 6% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSearchDnName(),
"ou=groups");
gsh 7% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapServerIdName(),
"personLdap");
gsh 8% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSourceIdName(),
"pennperson");
gsh 10% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectAttributeName(), "hasMember");
# NOTE: hopefully you can use subjectId instead, it will improve the performance a LOT!
gsh 11% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectIdTypeName(), "subjectIdentifier");
gsh 11% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapAndGroupsName(), "school:community:employee");
gsh 12% group = GroupFinder.findByName(grouperSession, "someStem3:myLdapGroup3");
gsh 13% loaderRunOneJob(group);
loader ran successfully, inserted 3 memberships, deleted 0 memberships, total membership count: 3
gsh 14% getMembers("someStem3:myLdapGroup3");
member: id='22345678' type='person' source='pennperson' uuid='360802a1bdf341859109c086ffe79022'
member: id='33456789' type='person' source='pennperson' uuid='5dd1fc0431214a6fa53bf3cb7790d5ea'
member: id='44567890' type='person' source='pennperson' uuid='8b26f3fb43da4661946282227580d5be'
gsh 5% delMember("someStem3:myLdapGroup3", "22345678");
true
gsh 16% addMember("someStem3:myLdapGroup3", "GrouperSystem");
true
gsh 17% loaderRunOneJob(group);

loader ran successfully, inserted 1 memberships, deleted 1 memberships, total membership count: 3
gsh 18%

```

LDAP_GROUP_LIST groupsLike test case

Add an extra group, which represents an orphan, assign a groupsLike, run the loader, see that group get its members removed, first do the common setup above, then you can configure and run the loader in GSH:

```

gsh 0% grouperSession = GrouperSession.startRootSession();
gsh 1% group = new GroupSave(grouperSession).assignName("anotherStem2:groupListLdapGroup2").
assignCreateParentStemsIfNotExist(true).save();
gsh 1% new GroupSave(grouperSession).assignName("anotherStem2:groups:someOrphan").
assignCreateParentStemsIfNotExist(true).save();
gsh 1% addMember("anotherStem2:groups:someOrphan", "22345678");
gsh 1% addMember("anotherStem2:groups:someOrphan", "33456789");
gsh 2% attributeAssign = group.getAttributeDelegate().assignAttribute(LoaderLdapUtils.
grouperLoaderLdapAttributeDefName()).getAttributeAssign();
#in case you need to retrieve again, here is an example
gsh 2% attributeAssign = group.getAttributeDelegate().retrieveAssignment(null, LoaderLdapUtils.
grouperLoaderLdapAttributeDefName(), false, true);
gsh 3% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapTypeName(),
"LDAP_GROUP_LIST");
gsh 4% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapFilterName(),
"(|(cn=test:testGroup)(cn=test:ldapTesting:test1))");
# every minute so it is easy to test
gsh 5% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapQuartzCronName(), "0 * * * * ?");
gsh 6% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSearchDnName(),
"ou=groups");
gsh 7% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapServerIdName(),
"personLdap");
gsh 8% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSourceIdName(),
"pennperson");
gsh 10% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectAttributeName(), "hasMember");
# NOTE: hopefully you can use subjectId instead, it will improve the performance a LOT!
gsh 11% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectIdTypeName(), "subjectIdentifier");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapExtraAttributesName(), "cn");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapGroupNameExpressionName(), "groups:${groupAttributes['cn']}");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapGroupsLikeName(), "anotherStem2:groups:%");
gsh 13% group = GroupFinder.findByName(grouperSession, "anotherStem2:groupListLdapGroup2");
gsh 13% loaderRunOneJob(group);
loader ran successfully, inserted 5 memberships, deleted 2 memberships, total membership count: 5
gsh 14% getGroups("anotherStem2")
group: name='anotherStem2:groups:test:testGroup' displayName='anotherStem2:groups:test:testGroup'
uuid='84cf356b1f694594971439ade9c32ce8'
group: name='anotherStem2:groupListLdapGroup' displayName='anotherStem2:groupListLdapGroup'
uuid='e42924f906ff4c0cb97bc8766338fea4'
group: name='anotherStem2:groups:test:ldapTesting:test1' displayName='anotherStem2:groups:test:ldapTesting:
test1' uuid='e767dcdf3d7340c38986ed4cdab30c44'
gsh 15% getGroups("anotherStem2:groups:someOrphan");
gsh 15% getMembers("anotherStem2:groups:test:testGroup");
member: id='22345678' type='person' source='pennperson' uuid='360802a1bdf341859109c086ffe79022'
member: id='33456789' type='person' source='pennperson' uuid='5dd1fc0431214a6fa53bf3cb7790d5ea'
member: id='44567890' type='person' source='pennperson' uuid='8b26f3fb43da4661946282227580d5be'
member: id='12345678' type='person' source='pennperson' uuid='db43860f64004ec295129cde994a450d'
gsh 15% getMembers("anotherStem2:groups:test:ldapTesting:test1");
member: id='10000000' type='person' source='pennperson' uuid='ea9b420cca1f43b1a1cb8b682cb3624a'
gsh 15% delMember("anotherStem2:groups:test:testGroup", "22345678");
true
gsh 16% addMember("anotherStem2:groups:test:ldapTesting:test1", "GrouperSystem");
true
gsh 17% loaderRunOneJob(group);

loader ran successfully, inserted 1 memberships, deleted 1 memberships, total membership count: 5
gsh 18%

```

LDAP_SIMPLE error unresolvable test case

First do the common setup above, then you can configure and run the loader in GSH. Delete a couple of records from the source via the DB

```
DELETE FROM person_source_v WHERE penn_id IN ('44567890', '12345678');
COMMIT;
```

```
gsh 0% grouperSession = GrouperSession.startRootSession();
gsh 1% group = new GroupSave(grouperSession).assignName("someStem4:myLdapGroup4").
assignCreateParentStemsIfNotExist(true).save();
gsh 2% attributeAssign = group.getAttributeDelegate().assignAttribute(LoaderLdapUtils.
grouperLoaderLdapAttributeDefName()).getAttributeAssign();
#in case you need to retrieve again, here is an example
gsh 2% attributeAssign = group.getAttributeDelegate().retrieveAssignment(null, LoaderLdapUtils.
grouperLoaderLdapAttributeDefName(), false, true);
gsh 3% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapTypeName(),
"LDAP_SIMPLE");
gsh 4% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapFilterName(),
"(|(cn=test:testGroup)(cn=test:ldaptesting:test1))");
# every minute so it is easy to test
gsh 5% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapQuartzCronName(), "0 * * * * ?");
gsh 6% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSearchDnName(),
"ou=groups");
gsh 7% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapServerIdName(),
"personLdap");
gsh 8% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSourceIdName(),
"pennperson");
gsh 10% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectAttributeName(), "hasMember");
# NOTE: hopefully you can use subjectId instead, it will improve the performance a LOT!
gsh 11% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectIdTypeName(), "subjectIdentifier");
gsh 12% group = GroupFinder.findByName(grouperSession, "someStem4:myLdapGroup4");
gsh 13% loaderRunOneJob(group);
2011-10-09 09:59:11,369: [main] ERROR GrouperLoaderResultSet$Row.getSubject(1102) - - Problem with
subjectIdentifier: mchyzer, subjectSourceId: pennperson, in jobName: LDAP_SIMPLE__someStem4:
myLdapGroup4__12b55951f4f34556b7b8ecd65b763d32
edu.internet2.middleware.subject.SubjectNotFoundException: Subject not found: select penn_id,name,description,
description_lower,pennname,email from person_source_v where pennname = ? or penn_id = ?, mchyzer,mchyzer
at edu.internet2.middleware.subject.provider.JDBCSourceAdapter2.search(JDBCSourceAdapter2.java:623)
loader ran successfully, inserted 3 memberships, deleted 0 memberships, total membership count: 3

# NOTE: the status of the job in the SQL table grouper_loader_log is: SUBJECT_PROBLEMS

gsh 14% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapErrorUnresolvableName(), "false");
gsh 14% loaderRunOneJob(group);

# NOTE: the status of the job in the SQL table grouper_loader_log is: SUCCESS, but still have as unresolvable
subject count of 2

loader ran successfully, inserted 0 memberships, deleted 0 memberships, total membership count: 3
gsh 14% getMembers("someStem4:myLdapGroup4");
member: id='22345678' type='person' source='pennperson' uuid='360802a1bdf341859109c086ffe79022'
member: id='33456789' type='person' source='pennperson' uuid='5dd1fc0431214a6fa53bf3cb7790d5ea'
member: id='100000000' type='person' source='pennperson' uuid='ea9b420cca1f43b1a1cb8b682cb3624a'
gsh 5% delMember("someStem4:myLdapGroup4", "22345678");
true
gsh 16% addMember("someStem4:myLdapGroup4", "GrouperSystem");
true
gsh 17% loaderRunOneJob(group);

loader ran successfully, inserted 1 memberships, deleted 1 memberships, total membership count: 3
gsh 18%
```

Add these back:

```
INSERT INTO `person_source_v`(`penn_id`,`name`,`description`,`pennname`,`description_lower`) VALUES
('44567890','Chris Hyzer','Chris Hyzer','mchyzer','chris hyzer, mchyzer, 44567890'),
('12345678','Bryan Hall','Bryan Hall','bwh','bryan hall, bwh, 12345678')
;
COMMIT;
```

LDAP_GROUPS_FROM_ATTRIBUTES group name, group display name, group description test case

I dont have a multi-valued user attribute, I will just do an inverted loader job from the above... the group ldap objects are the subjects (by identifier), and the hasmember multi-valued attributes will be the groups :)

```
gsh 0% grouperSession = GrouperSession.startRootSession();
gsh 1% group = new GroupSave(grouperSession).assignName("yetAnotherStem3:groupsFromAttributesLdapGroup3").
assignCreateParentStemsIfNotExist(true).save();
gsh 2% new GroupSave(grouperSession).assignName("test:testGroup").assignCreateParentStemsIfNotExist(true).
save();
gsh 2% new GroupSave(grouperSession).assignName("test:testGroup2").assignCreateParentStemsIfNotExist(true).
save();

gsh 2% attributeAssign = group.getAttributeDelegate().assignAttribute(LoaderLdapUtils.
grouperLoaderLdapAttributeDefName()).getAttributeAssign();
#in case you need to retrieve again, here is an example
gsh 2% attributeAssign = group.getAttributeDelegate().retrieveAssignment(null, LoaderLdapUtils.
grouperLoaderLdapAttributeDefName(), false, true);
gsh 3% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapTypeName(),
"LDAP_GROUPS_FROM_ATTRIBUTES");
gsh 4% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapFilterName(),
"(|(cn=test:testGroup)(cn=test:testGroup2))");
# every minute so it is easy to test
gsh 5% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapQuartzCronName(), "0 * * * * ?");
gsh 6% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSearchDnName(),
"ou=groups");
gsh 7% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapServerIdName(),
"personLdap");
gsh 8% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSourceIdName(),
"g:gsa");
gsh 8% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapGroupAttributeName(), "hasmember");

# NOTE: hopefully you can use subjectId instead, it will improve the performance a LOT!
gsh 11% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectIdTypeName(), "subjectIdentifier");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapGroupNameExpressionName(), "groups:${groupAttribute}");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapGroupDisplayNameExpressionName(), "Loaded groups:${grouperUtil.capitalize(groupAttribute)}");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapGroupDescriptionExpressionName(), "Loaded group from ldap attribute value hasMember:
${groupAttribute}");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectExpressionName(), "${subjectAttributes['cn']}");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapExtraAttributesName(), "cn");
gsh 13% group = GroupFinder.findByName(grouperSession, "yetAnotherStem3:groupsFromAttributesLdapGroup3");
gsh 13% loaderRunOneJob(group);
loader ran successfully, inserted 8 memberships, deleted 0 memberships, total membership count: 8
gsh 14% for (theGroup : getGroups("yetAnotherStem3:groups")) { print(theGroup.getName());print(theGroup.
getDisplayName());print(theGroup.getDescription()); }
yetAnotherStem3:groupsFromAttributesLdapGroup3
yetAnotherStem3:groupsFromAttributesLdapGroup3

yetAnotherStem3:groups:choate
yetAnotherStem3:Loaded groups:Choate
Loaded group from ldap attribute value hasMember: choate
yetAnotherStem3:groups:taberkow
```

```

yetAnotherStem3:Loaded groups:Taberkow
Loaded group from ldap attribute value hasMember: taberkow
yetAnotherStem3:groups:mchzyer
yetAnotherStem3:Loaded groups:Mchzyer
Loaded group from ldap attribute value hasMember: mchzyer
yetAnotherStem3:groups:bwh
yetAnotherStem3:Loaded groups:Bwh
Loaded group from ldap attribute value hasMember: bwh
yetAnotherStem3:groups:harveycg
yetAnotherStem3:Loaded groups:Harveycg
Loaded group from ldap attribute value hasMember: harveycg
yetAnotherStem3:groups:convery
yetAnotherStem3:Loaded groups:Convery
Loaded group from ldap attribute value hasMember: convery
gsh 15% getMembers("yetAnotherStem3:groups:choate");
member: id='22345678' type='person' source='pennperson' uuid='360802a1bdf341859109c086ffe79022'
member: id='33456789' type='person' source='pennperson' uuid='5dd1fc0431214a6fa53bf3cb7790d5ea'
member: id='44567890' type='person' source='pennperson' uuid='8b26f3fb43da4661946282227580d5be'
member: id='12345678' type='person' source='pennperson' uuid='db43860f64004ec295129cde994a450d'
gsh 15% for (theGroup : getMembers("yetAnotherStem3:groups:choate")) {print(theGroup.getName());}
test:testGroup
test:testGroup2
gsh 15% delMember("yetAnotherStem3:groups:choate", "test:testGroup");
true
gsh 16% addMember("yetAnotherStem3:groups:choate", "GrouperSystem");
true
gsh 17% loaderRunOneJob(group);

loader ran successfully, inserted 1 memberships, deleted 1 memberships, total membership count: 8
gsh 18%

```

LDAP_GROUPS_FROM_ATTRIBUTES with attribute filter expression (JEXL) (2.1.4+)

```

Grouper note: auto-created attributeDefName: etc:attribute:loaderLdap:grouperLoaderLdapAttributeFilterExpression
2013-05-13 18:25:06,709: [main] WARN GrouperCheckConfig.checkAttribute(1554) - - auto-created attributeDefName:
etc:attribute:loaderLdap:grouperLoaderLdapAttributeFilterExpression
Type help() for instructions
gsh 0% grouperSession = GrouperSession.startRootSession();
edu.internet2.middleware.grouper.GrouperSession: 1ccf09dc2542477ba53c09a7a5f0ebb0,'GrouperSystem','application'
gsh 1% group = GroupFinder.findByName(grouperSession, "yetAnotherStem:groupsFromAttributesLdapGroup");
gsh 2% attributeAssign = group.getAttributeDelegate().retrieveAssignment(null, LoaderLdapUtils.
grouperLoaderLdapAttributeDefName(), false, true);
gsh 3% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapAttributeFilterExpressionName(), "${attributeValue == 'whatever'}");
gsh 4% loaderRunOneJob(group);
2013-05-13 18:26:32,520: [main] DEBUG GrouperLoaderResultSet$2.callback(854) - - Attribute 'convery' is allowed
to be used based on expression? false, '${attributeValue == 'whatever'}', note the attributeValue is in a
variable called attributeValue
gsh 5% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapAttributeFilterExpressionName(), "${attributeValue == 'convery'}");
gsh 6% loaderRunOneJob(group);
2013-05-13 18:27:38,355: [main] DEBUG GrouperLoaderResultSet$2.callback(854) - - Attribute 'convery' is allowed
to be used based on expression? true, '${attributeValue == 'mchzyer' || attributeValue == 'bwh'}', note the
attributeValue is in a variable called attributeValue
loader ran successfully, inserted 1 memberships, deleted 0 memberships, total membership count: 1
gsh 7%

```

LDAP_GROUP_LIST privileges test case

Create some privilege groups, assign them to each created group. First do the common setup above, then you can configure and run the loader in GSH:

```

gsh 0% grouperSession = GrouperSession.startRootSession();
gsh 1% group = new GroupSave(grouperSession).assignName("anotherStem3:groupListLdapGroup3").
assignCreateParentStemsIfNotExist(true).save();
gsh 1% new GroupSave(grouperSession).assignName("anotherStem3:privs:readers").assignCreateParentStemsIfNotExist
(true).save();

```

```

gsh 1% new GroupSave(grouperSession).assignName("anotherStem3:privs:admins").assignCreateParentStemsIfNotExist
(true).save();
gsh 1% new GroupSave(grouperSession).assignName("anotherStem3:privs:viewers").assignCreateParentStemsIfNotExist
(true).save();
gsh 1% new GroupSave(grouperSession).assignName("anotherStem3:privs:updaters").assignCreateParentStemsIfNotExist
(true).save();
gsh 1% new GroupSave(grouperSession).assignName("anotherStem3:privs:optins").assignCreateParentStemsIfNotExist
(true).save();
gsh 1% new GroupSave(grouperSession).assignName("anotherStem3:privs:optouts").assignCreateParentStemsIfNotExist
(true).save();
gsh 2% attributeAssign = group.getAttributeDelegate().assignAttribute(LoaderLdapUtils.
grouperLoaderLdapAttributeDefName()).getAttributeAssign();
#in case you need to retrieve again, here is an example
gsh 2% attributeAssign = group.getAttributeDelegate().retrieveAssignment(null, LoaderLdapUtils.
grouperLoaderLdapAttributeDefName(), false, true);
gsh 3% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapTypeName(),
"LDAP_GROUP_LIST");
gsh 4% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapFilterName(),
"(|(cn=test:testGroup)(cn=test:ldapTesting:test1))");
# every minute so it is easy to test
gsh 5% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapQuartzCronName(), "0 * * * * ?");
gsh 6% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSearchDnName(),
"ou=groups");
gsh 7% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapServerIdName(),
"personLdap");
gsh 8% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapSourceIdName(),
"pennperson");
gsh 10% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectAttributeName(), "hasMember");
# NOTE: hopefully you can use subjectId instead, it will improve the performance a LOT!
gsh 11% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapSubjectIdTypeName(), "subjectIdentifier");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapExtraAttributesName(), "cn");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.
grouperLoaderLdapGroupNameExpressionName(), "groups:${groupAttributes['cn']}");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapReadersName(),
"anotherStem3:privs:readers");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapUpdatersName(),
"anotherStem3:privs:updaters");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapAdminsName(),
"anotherStem3:privs:admins");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapViewersName(),
"anotherStem3:privs:viewers");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapOptinsName(),
"anotherStem3:privs:optins");
gsh 12% attributeAssign.getAttributeValueDelegate().assignValue(LoaderLdapUtils.grouperLoaderLdapOptoutsName(),
"anotherStem3:privs:optouts");
gsh 13% group = GroupFinder.findByName(grouperSession, "anotherStem3:groupListLdapGroup3");
gsh 13% loaderRunOneJob(group);
loader ran successfully, inserted 5 memberships, deleted 0 memberships, total membership count: 5
gsh 14% getGroups("anotherStem3");
group: name='anotherStem3:privs:updaters' displayName='anotherStem3:privs:updaters'
uuid='00e6678238d54d02bc4af7886977e800'
group: name='anotherStem3:privs:optins' displayName='anotherStem3:privs:optins'
uuid='096c5038fdf94bf2b84750f795962e43'
group: name='yetAnotherStem3:groupsFromAttributesLdapGroup3' displayName='yetAnotherStem3:
groupsFromAttributesLdapGroup3' uuid='177991996a2343b489b7e20a287e484d'
group: name='yetAnotherStem3:groups:choate' displayName='yetAnotherStem3:Loaded groups:Choate'
uuid='20033d406c3540f6b21a8767739253fb'
group: name='anotherStem3:privs:optouts' displayName='anotherStem3:privs:optouts'
uuid='39063c2fc7c24dbcbbaa64e2576b60100'
group: name='yetAnotherStem3:groups:taberkow' displayName='yetAnotherStem3:Loaded groups:Taberkow'
uuid='4423e84c1c2942078104586128aa30e8'
group: name='anotherStem3:groups:test:ldapTesting:test1' displayName='anotherStem3:groups:test:ldapTesting:
test1' uuid='52e75327d1144e46b8d8ca3e9bd12601'
group: name='yetAnotherStem3:groups:mchyzer' displayName='yetAnotherStem3:Loaded groups:Mchyzer'
uuid='8403596a647f42edbd6989a30b406af0'
group: name='anotherStem3:privs:admins' displayName='anotherStem3:privs:admins'
uuid='9a398c0a11504838a1f5cbfbc025e9d9'

```

```

group: name='yetAnotherStem3:groups:bwh' displayName='yetAnotherStem3:Loaded groups:Bwh'
uuid='d195aabfc0d344bdaef96730844f6d7c'
group: name='yetAnotherStem3:groups:harveycg' displayName='yetAnotherStem3:Loaded groups:Harveycg'
uuid='d1a6f2cd7bc746989ed5ae44508b9795'
group: name='anotherStem3:privs:readers' displayName='anotherStem3:privs:readers'
uuid='d366d9a478fe4db5963f112a4dd2f78d'
group: name='anotherStem3:groups:test:testGroup' displayName='anotherStem3:groups:test:testGroup'
uuid='d8cb545b8d424e8e8733966d240b9225'
group: name='anotherStem3:groupListLdapGroup3' displayName='anotherStem3:groupListLdapGroup3'
uuid='eb052fa06cf34a399e9486785e9996b1'
group: name='yetAnotherStem3:groups:convery' displayName='yetAnotherStem3:Loaded groups:Convery'
uuid='f47c07b37d5d4425a93091538594fa7a'
group: name='anotherStem3:privs:viewers' displayName='anotherStem3:privs:viewers'
uuid='fb19a041f5c040b08b00613e563d6bc8'
gsh 15% getMembers("anotherStem3:groups:test:testGroup");
member: id='22345678' type='person' source='pennperson' uuid='360802a1bdf341859109c086ffe79022'
member: id='33456789' type='person' source='pennperson' uuid='5dd1fc0431214a6fa53bf3cb7790d5ea'
member: id='44567890' type='person' source='pennperson' uuid='8b26f3fb43da4661946282227580d5be'
member: id='12345678' type='person' source='pennperson' uuid='db43860f64004ec295129cde994a450d'
gsh 15% getMembers("anotherStem3:groups:test:ldapTesting:test1");
member: id='10000000' type='person' source='pennperson' uuid='ea9b420cca1f43b1a1cb8b682cb3624a'

gsh 15% hasPriv("anotherStem3:groups:test:testGroup", "anotherStem3:privs:readers", AccessPrivilege.ADMIN);
false
gsh 15% hasPriv("anotherStem3:groups:test:testGroup", "anotherStem3:privs:readers", AccessPrivilege.READ);
true
gsh 15% hasPriv("anotherStem3:groups:test:testGroup", "anotherStem3:privs:admins", AccessPrivilege.ADMIN);
true
gsh 15% hasPriv("anotherStem3:groups:test:testGroup", "anotherStem3:privs:admins", AccessPrivilege.READ);
true
gsh 15% hasPriv("anotherStem3:groups:test:testGroup", "anotherStem3:privs:updaters", AccessPrivilege.UPDATE);
true
gsh 15% hasPriv("anotherStem3:groups:test:testGroup", "anotherStem3:privs:updaters", AccessPrivilege.ADMIN);
false
gsh 15% hasPriv("anotherStem3:groups:test:testGroup", "anotherStem3:privs:viewers", AccessPrivilege.ADMIN);
false
gsh 15% hasPriv("anotherStem3:groups:test:testGroup", "anotherStem3:privs:viewers", AccessPrivilege.VIEW);
true
gsh 15% hasPriv("anotherStem3:groups:test:testGroup", "anotherStem3:privs:optins", AccessPrivilege.OPTIN);
true
gsh 15% hasPriv("anotherStem3:groups:test:testGroup", "anotherStem3:privs:optins", AccessPrivilege.OPTOUT);
false
gsh 15% hasPriv("anotherStem3:groups:test:testGroup", "anotherStem3:privs:optouts", AccessPrivilege.OPTOUT);
true
gsh 15% hasPriv("anotherStem3:groups:test:testGroup", "anotherStem3:privs:optouts", AccessPrivilege.OPTIN);
false
gsh 15% delMember("anotherStem:groups:test:testGroup", "22345678");
true
gsh 16% addMember("anotherStem:groups:test:ldapTesting:test1", "GrouperSystem");
true
gsh 17% loaderRunOneJob(group);

loader ran successfully, inserted 1 memberships, deleted 1 memberships, total membership count: 5
gsh 18%

```

Diagnostics

Go to this URL in grouper WS: <http://server.institution.edu/grouper-ws/status?diagnosticType=all>, see the LDAP jobs there, checked to make sure ok for tool like nagios

```

Server: mchyszer-PC, grouperVersion: 2.1.0, up since: 2011/10/10 02:35, 0 requests
SUCCESS memoryTest: Allocating 100000 bytes to an array to make sure not out of memory (25ms elapsed)
SUCCESS dbTest_grouper: Retrieved object from cache (25ms elapsed)
SUCCESS source_g:gsa: Source checked successfully recently (25ms elapsed)
SUCCESS source_pennperson: Source checked successfully recently (25ms elapsed)
SUCCESS source_jdbc: Source checked successfully recently (25ms elapsed)
SUCCESS source_g:isa: Source checked successfully recently (25ms elapsed)
SUCCESS loader_CHANGE_LOG_changeLogTempToChangeLog: Not checking, there was a success from before: 2011/10/10
02:34:50.000, expecting one in the last 30 minutes (25ms elapsed)
SUCCESS loader_MAINTENANCE_cleanLogs: Not checking, there was a success from before: 2011/10/10 01:27:09.000,
expecting one in the last 1500 minutes (25ms elapsed)
SUCCESS loader_CHANGE_LOG_consumer_syncGroups: Not checking, there was a success from before: 2011/10/10 02:35:
00.000, expecting one in the last 30 minutes (25ms elapsed)
SUCCESS loader_CHANGE_LOG_consumer_grouperRules: Not checking, there was a success from before: 2011/10/10 02:
35:02.000, expecting one in the last 30 minutes (26ms elapsed)
SUCCESS loader_LDAP_GROUP_LIST__anotherStem2:groupListLdapGroup2__61c103adcce84584991c2e1e9ae5e280: Not
checking, there was a success from before: 2011/10/09 09:38:35.000, expecting one in the last 1500 minutes
(26ms elapsed)
SUCCESS loader_LDAP_SIMPLE__someStem3:myLdapGroup3__12b55951f4f34556b7b8ecd65b763d32: Not checking, there was a
success from before: 2011/10/10 02:35:00.000, expecting one in the last 1500 minutes (26ms elapsed)
SUCCESS loader_LDAP_GROUPS_FROM_ATTRIBUTES__yetAnotherStem2:
groupsFromAttributesLdapGroup2__3ae9ed78b74c48a081f8f02ca96f699e: Not checking, there was a success from
before: 2011/10/10 01:54:04.000, expecting one in the last 1500 minutes (26ms elapsed)
SUCCESS loader_LDAP_GROUP_LIST__anotherStem:groupListLdapGroup__e42924f906ff4c0cb97bc8766338fea4: Not checking,
there was a success from before: 2011/10/10 01:55:40.000, expecting one in the last 1500 minutes (26ms elapsed)
SUCCESS loader_LDAP_SIMPLE__someStem4:myLdapGroup4__ab8cd26fe11e4e19ad1618806d4740cc: Not checking, there was a
success from before: 2011/10/10 02:35:00.000, expecting one in the last 1500 minutes (26ms elapsed)
SUCCESS loader_LDAP_SIMPLE__someStem2:myLdapGroup2__875108a266f54f70baf1d39b27fa5745: Not checking, there was a
success from before: 2011/10/10 02:35:00.000, expecting one in the last 1500 minutes (26ms elapsed)
SUCCESS loader_LDAP_GROUP_LIST__anotherStem3:groupListLdapGroup3__eb052fa06cf34a399e9486785e9996b1: Not
checking, there was a success from before: 2011/10/09 14:21:39.000, expecting one in the last 1500 minutes
(26ms elapsed)
SUCCESS loader_LDAP_GROUPS_FROM_ATTRIBUTES__yetAnotherStem:
groupsFromAttributesLdapGroup__9dd5e5eca4f04400965ed3bc0b986ec2: Not checking, there was a success from before:
2011/10/10 01:56:22.000, expecting one in the last 1500 minutes (27ms elapsed)
SUCCESS loader_LDAP_SIMPLE__someStem:myLdapGroup__aa8f3a245d1947509d347fee0f6a80b2: Not checking, there was a
success from before: 2011/10/10 02:35:00.000, expecting one in the last 1500 minutes (27ms elapsed)
SUCCESS loader_LDAP_GROUPS_FROM_ATTRIBUTES__yetAnotherStem3:
groupsFromAttributesLdapGroup3__177991996a2343b489b7e20a287e484d: Not checking, there was a success from
before: 2011/10/09 14:05:07.000, expecting one in the last 1500 minutes (27ms elapsed)

Diagnostics errors since start: 0 (27ms elapsed)

```

Example of converting DN to subjectId or Group name (institution specific)

In the grouper-loader.properties, add the class

```
loader.ldap.el.classes = edu.internet2.middleware.grouper.app.loader.ldap.LdapGroupUserConverter
```

Add the [LdapGroupUserConverter.jar](#) to the classpath (e.g. to lib/custom)

In the grouper-loader.properties, add the class

```
loader.ldap.el.classes = ldapGroupUserConverter.LdapGroupUserConverter
```

Set the Grouper loader LDAP subject expression attribute to `${ldapGroupUserConverter.convertDntoSubjectIdOrIdentifier(subjectId)}`

Unset the subject source id

If the subjectId is a subjectId, then make sure Grouper loader LDAP subject ID type is "subjectIdOrIdentifier". If it is a subjectIdentifier (more common), then you can set it as subjectIdentifier.

Log the conversions with this in log4j2.xml or in grouper.properties

```
grouper.logger.LdapGroupUserConverter.name = edu.internet2.middleware.grouper.app.loader.ldap.  
LdapGroupUserConverter  
grouper.logger.LdapGroupUserConverter.level = debug
```

~~Log the conversions with this in log4j.properties~~

```
log4j.logger.LdapGroupUserConverter.LdapGroupUserConverter = DEBUG
```