v2.0 Release Notes

Release Notes for Grouper v2.0

Grouper v2.0.3 fixes a serious SQL problem with 2.0.2

Grouper v2.0.2 fixes a couple dozen issues including making subject searches more efficient and some UI fixes

Grouper v2.0.1 fixes several issues

Grouper v2.0.0 includes 47 fixes and improvements over v1.6.3. See the full list.

New Features

Rules	Similar to Grouper Hooks, but instead of Java logic, built in actions or expression language scripts can be executed
External subjects	If your Identity Management System does not support external users (e.g. via EPPN), then Grouper can manage that with self registration and or invitations which will can provision memberships
Syncing groupers	A group in one Grouper can be sync'ed with a group in another Grouper. For instance if two institutions want to share a group of subjects but store them in their own Grouper
Attribute and Permissions UI	User interface to define, view, and assign attributes and permissions in Grouper. The attributes can be assigned to many types of Grouper objects including Groups, Folders, Members, Memberships, etc. The permissions are used as a central permissions management system for other applications at your institution
Grouper-Atlassian connector	If you cannot connect Atlassian applications (e.g. Jira, Confluence) to your Grouper managed LDAP, then you can use this connector which used Grouper Web Services to manage your Atlassian groups and person information
Permissions Allow /disallow	A permission assignment can be an allow or disallow (to filter out allows inherited from another assignment)
Permission limits	A run-time decision can be applied to immediate permission allows so that context environment variables can change an allow to a disallow. e.g. permissions are only allowed at a certain time of day or from a certain IP address. Grouper can calculate this on the server or the client can get the limits and calculate them.
Web service versioning	Grouper 2.0 web servers will accept clients coded against Grouper 1.6 or previous WS API's
Point in Time Audit	This allows you to query the state of Grouper at a point in time in the past or a date range in the past. You can query for memberships, privileges and permissions.

For more information about upcoming plans, see the Grouper+Product+Roadmap.

Improvements & Fixes

Member Search and Sort	Additional data is now stored about subjects in Grouper. This allows you to sort a list of members and search a list of members without having to go to the subject source to query attributes for each subject in the list that you would then use for the sort or search operation.
ldappcng caching (performance)	The SPMLDataConnector supports caching similar to other Shibboleth DataConnectors
Notification improvements	Additional notifications are available now for permissions and the attribute framework.

Many other fixes and improvements were also made to all components of the Grouper Toolkit: Grouper API, Administrative & Lite UIs, Grouper Web Services, Grouper Client, Grouper Shell, Grouper Loader, Ldappc, Ldappc-ng, and the Subject API.

Upgrading from Grouper v1.6

The following instructions describe how you can upgrade to 2.0 from 1.6. To give you an idea of how long the database upgrade may take, I performed a test upgrade on an Oracle database with 125,400 groups, 105,710 stems, 1,067,124 memberships, and 128,328 members. The actual database upgrade steps (Step 7, Step 10, Step 11) took the following amount of time:

- Step 7 (generate SQL script): 3 minutes
- Step 10 (run SQL script): 13 minutes
- Step 11 (update grouper_members table): 50 minutes

Your time will vary depending on several factors such as the type of database you are using, how well it is tuned, how fast your subject source responds to queries, etc... Also, if you really wanted to, you can perform Step 11 after giving your users access to the Grouper UI/WS, etc again but membership results in the UI may not sort properly until that step is done.

1. You should get v2.0 versions of the Grouper API, Grouper UI, Grouper WS, Grouper Daemon, etc. You will need to merge configuration files and JARs. See the change log for more information. The rest of this document focuses on upgrading the database.
2. First you may want to analyze your tables to help speed up the upgrade. Analyze your tables. At minimum, be sure to analyze grouper_members, grouper_group_set, grouper_memberships, grouper_groups, and grouper_stems.
3. You may need to increase tablespace for your schema since the upgrade will add point in time auditing.
4. Once you prevent users from making updates to your Grouper instance, run the changeLogTempToChangeLog daemon to clear out the temp changelog using the v1.6 API. Here's an example using GSH.

   ```
   gsh 0% loaderRunOneJob("CHANGE_LOG_changeLogTempToChangeLog")
   ```

5. If you are not currently using the change log for notifications, then you can instead just clear the temp change log.

   ```
   delete from grouper_change_log_entry_temp; commit;
   ```

6. Before performing any upgrade steps, export your Grouper registry. Options include performing a database backup or using the XML Export utility in Grouper.
7. Using the 2.0 API, perform a registry check using GSH to create an SQL file that will contain the DDL to update your database. To do this, run: gsh -registry -check For instance..

```
$ ./bin/gsh.sh -registry -check
Using GROUPER_HOME: /srv/grouper
Using GROUPER_CONF: /srv/grouper/conf
Using JAVA: java
using MEMORY: 64m-512m
Grouper starting up: version: 2.0.0, build date: 2011/07/30 12:40:43, env: <no label configured>
grouper.properties read from: /srv/grouper/conf/grouper.properties
Grouper current directory is: /srv/grouper
log4j.properties read from:   /srv/grouper/conf/log4j.properties
Grouper is logging to file:   /srv/grouper/logs/grouper_error.log, at min level WARN for package: edu.
internet2.middleware.grouper, based on log4j.properties
grouper.hibernate.properties: /srv/grouper/conf/grouper.hibernate.properties
grouper.hibernate.properties: ims@jdbc:oracle:thin:@imstst-db.oit.duke.edu:1668:IMSTST
sources.xml read from:        /srv/grouper/conf/sources.xml
sources.xml groupersource id: g:gsa
sources.xml jdbc source id:   jdbc: GrouperJdbcConnectionProvider
(note, might need to type in your response multiple times (Java stdin is flaky))
(note, you can allow and deny db urls and users in the grouper.properties)
Are you sure you want to schemaexport all tables (dropThenCreate=F,writeAndRunScript=F) in db user
'ims', db url 'jdbc:oracle:thin:@imstst-db.oit.duke.edu:1668:IMSTST'? (y|n):
y
Continuing...
Grouper ddl object type 'Grouper' has dbVersion: 23 and java version: 25
Grouper database schema DDL requires updates
(should run script manually and carefully, in sections, verify data before drop statements, backup
/export important data before starting, follow change log on confluence, dont run exact same script in
multiple envs - generate a new one for each env),
script file is:
/srv/grouper/ddlScripts/grouperDdl_20110730_13_40_54_757.sql
Note: this script was not executed due to option passed in
To run script via gsh, carefully review it, then run this:
gsh -registry -runsqlfile /srv/grouper/ddlScripts/grouperDdl_20110730_13_40_54_757.sql
```

8. In this example above, an SQL script called /srv/grouper/ddlScripts/grouperDdl_20110730_13_40_54_757.sql was created.
9. Review the script to make sure it looks okay. The script will be dropping and recreating the table GROUPER_PIT_ATTR_ASSN_VALUE. It will also drop and recreate views, constraints, and some indexes. And it will drop the GROUPER_FLAT_* tables. The tables GROUPER_ATTRIBUTE_ASSIGN, GROUPER_GROUPS, and GROUPER_MEMBERS will have new columns added.
    a. If using postgres, you should see foreign keys being dropped at the top of the script. If not, try setting the ddlutils.schema grouper. properties setting and run again. If you still dont see foreign keys being dropped at the top of the script, manually drop all foreign keys before running the script.
    b. If using postgres or hsql, you should backup any non grouper views that depend on Grouper views, run the grouper script (which deletes those views due to drop view cascade), and then you should recreate those non grouper views.
10. If you are okay with the SQL script, execute using GSH again. To do this, run: gsh -registry -runsqlfile /path/to/sql/file.sql For instance..

```
$ ./bin/gsh.sh -registry -runsqlfile ddlScripts/grouperDdl_20110730_13_40_54_757.sql
Using GROUPER_HOME: /srv/grouper
Using GROUPER_CONF: /srv/grouper/conf
Using JAVA: java
using MEMORY: 64m-512m
(note, might need to type in your response multiple times (Java stdin is flaky))
(note, you can allow and deny db urls and users in the grouper.properties)
Are you sure you want to run the sql file in db user 'ims', db url 'jdbc:oracle:thin:@imstst-db.oit.duke.
edu:1668:IMSTST'? (y|n):
y
Continuing...
Script was executed successfully

Grouper starting up: version: 2.0.0, build date: 2011/07/30 12:40:43, env: <no label configured>
grouper.properties read from: /srv/grouper/conf/grouper.properties
Grouper current directory is: /srv/grouper
log4j.properties read from:   /srv/grouper/conf/log4j.properties
Grouper is logging to file:   /srv/grouper/logs/grouper_error.log, at min level WARN for package: edu.
internet2.middleware.grouper, based on log4j.properties
grouper.hibernate.properties: /srv/grouper/conf/grouper.hibernate.properties
grouper.hibernate.properties: ims@jdbc:oracle:thin:@imstst-db.oit.duke.edu:1668:IMSTST
sources.xml read from:        /srv/grouper/conf/sources.xml
sources.xml groupersource id: g:gsa
sources.xml jdbc source id:   jdbc: GrouperJdbcConnectionProvider
```

11. Starting with v2.0, Grouper now stores member attributes that you can configure and use to sort and search a list of members. These attributes are populated in the member objects when the subjects are resolved in Grouper. Here is how you can resolve the subjects. Note that if you have a lot of groups or members, you may have to increase your JVM heap size before starting GSH.

```
$ ./bin/gsh.sh
Using GROUPER_HOME: /srv/grouper
Using GROUPER_CONF: /srv/grouper/conf
Using JAVA: java
using MEMORY: 64m-512m
Grouper starting up: version: 2.0.0, build date: 2011/07/30 12:40:43, env: <no label configured>
grouper.properties read from: /srv/grouper/conf/grouper.properties
Grouper current directory is: /srv/grouper
log4j.properties read from:   /srv/grouper/conf/log4j.properties
Grouper is logging to file:   /srv/grouper/logs/grouper_error.log, at min level WARN for package: edu.
internet2.middleware.grouper, based on log4j.properties
grouper.hibernate.properties: /srv/grouper/conf/grouper.hibernate.properties
grouper.hibernate.properties: ims@jdbc:oracle:thin:@imstst-db.oit.duke.edu:1668:IMSTST
sources.xml read from:        /srv/grouper/conf/sources.xml
sources.xml groupersource id: g:gsa
sources.xml jdbc source id:   jdbc: GrouperJdbcConnectionProvider
Grouper warning: jarfile mismatch, expecting name: 'subject.jar' size: 118749 manifest version: 1.6.0.
However the jar detected is: /srv/grouper/lib/grouper/subject.jar, name: subject.jar size: 147811
manifest version: 2.0.0
Grouper note: auto-created attributeDefName: etc:attribute:attrExternalSubjectInvite:
externalSubjectInvite
Grouper note: auto-created attributeDefName: etc:attribute:attrExternalSubjectInvite:
externalSubjectInviteExpireDate
Grouper note: auto-created attributeDefName: etc:attribute:attrExternalSubjectInvite:
externalSubjectInviteDate
Grouper note: auto-created attributeDefName: etc:attribute:attrExternalSubjectInvite:
externalSubjectEmailAddress
Grouper note: auto-created attributeDefName: etc:attribute:attrExternalSubjectInvite:
externalSubjectInviteGroupUuids
Grouper note: auto-created attributeDefName: etc:attribute:attrExternalSubjectInvite:
externalSubjectInviteMemberId
Grouper note: auto-created attributeDefName: etc:attribute:attrExternalSubjectInvite:
externalSubjectInviteUuid
Grouper note: auto-created attributeDefName: etc:attribute:attrExternalSubjectInvite:
externalSubjectInviteEmailWhenRegistered
Grouper note: auto-created attributeDefName: etc:attribute:attrExternalSubjectInvite:
externalSubjectInviteEmail
```

```
Grouper note: auto-created attributeDefName: etc:attribute:rules:rule
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleActAsSubjectId
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleActAsSubjectIdentifier
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleActAsSubjectSourceId
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleCheckType
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleCheckOwnerId
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleCheckOwnerName
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleCheckStemScope
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleCheckArg0
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleCheckArg1
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleIfOwnerId
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleIfOwnerName
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleIfConditionEl
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleIfConditionEnum
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleIfConditionEnumArg0
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleIfConditionEnumArg1
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleIfStemScope
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleThenEl
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleThenEnum
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleThenEnumArg0
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleThenEnumArg1
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleThenEnumArg2
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleValid
Grouper note: auto-created attributeDefName: etc:attribute:rules:ruleRunDaemon
Grouper note: auto-created attributeDefName: etc:attribute:permissionLimits:limitExpression
Grouper note: auto-created attributeDefName: etc:attribute:permissionLimits:limitIpOnNetworks
Grouper note: auto-created attributeDefName: etc:attribute:permissionLimits:limitIpOnNetworkRealm
Grouper note: auto-created attributeDefName: etc:attribute:permissionLimits:limitLabelsContain
Grouper note: auto-created attributeDefName: etc:attribute:permissionLimits:limitAmountLessThan
Grouper note: auto-created attributeDefName: etc:attribute:permissionLimits:limitAmountLessThanOrEqual
Grouper note: auto-created attributeDefName: etc:attribute:permissionLimits:limitWeekday9to5
Grouper note: auto-created attributeDefName: etc:attribute:attrLoader:attributeLoader
Grouper note: auto-created attributeDefName: etc:attribute:attrLoader:attributeLoaderType
Grouper note: auto-created attributeDefName: etc:attribute:attrLoader:attributeLoaderDbName
Grouper note: auto-created attributeDefName: etc:attribute:attrLoader:attributeLoaderScheduleType
Grouper note: auto-created attributeDefName: etc:attribute:attrLoader:attributeLoaderQuartzCron
Grouper note: auto-created attributeDefName: etc:attribute:attrLoader:attributeLoaderIntervalSeconds
Grouper note: auto-created attributeDefName: etc:attribute:attrLoader:attributeLoaderPriority
Grouper note: auto-created attributeDefName: etc:attribute:attrLoader:attributeLoaderAttrsLike
Grouper note: auto-created attributeDefName: etc:attribute:attrLoader:attributeLoaderAttrQuery
Grouper note: auto-created attributeDefName: etc:attribute:attrLoader:attributeLoaderAttrSetQuery
Grouper note: auto-created attributeDefName: etc:attribute:attrLoader:attributeLoaderActionQuery
Grouper note: auto-created attributeDefName: etc:attribute:attrLoader:attributeLoaderActionSetQuery
Type help() for instructions
gsh 0% // run USDU to resolve all the subjects with type=person
gsh 1% subject=SubjectFinder.findById("GrouperSystem")
subject: id='GrouperSystem' type='application' source='g:isa' name='GrouperSysAdmin'
gsh 2% session=GrouperSession.start(subject)
edu.internet2.middleware.grouper.GrouperSession:
8106bdad683d43f88bf24c8e683f6162,'GrouperSystem','application'
gsh 3% usdu()
usdu completed successfully
gsh 4% // resolve the groups
gsh 5% GrouperSession.startRootSession();
gsh 6% for (String g : HibernateSession.byHqlStatic().createQuery("select uuid from Group").listSet
(String.class)) { subj = SubjectFinder.findByIdAndSource(g, "g:gsa", true); GrouperDAOFactory.
getFactory().getMember().findBySubject(subj).updateMemberAttributes(subj, true); }
```

12. Analyze your tables. At minimum, be sure to analyze grouper_members, grouper_group_set, grouper_memberships, grouper_groups, grouper_stems, grouper_pit_members, grouper_pit_group_set, grouper_pit_memberships, grouper_pit_groups, and grouper_pit_stems.
13. Start the Grouper Loader.