v1.6 Release Notes

Release Notes for Grouper v1.6

Grouper v1.6.3 fixes several issues including a security fix

Grouper v1.6.2 fixes 3 issues

Grouper v1.6.1 fixes 26 issues including partial XML export.

Grouper v1.6.0 includes more than 60 fixes and improvements over v1.5.3. See the full list.

New Features

Web service enhancements	Web services now support the new attribute and permission frameworks, subject service, new options for existing operations.
Kuali Rice integration	Connector for Kuali Rice to communicate with Grouper, can delegate Group decisions to Grouper and facilitate workflow in Grouper actions.
New Ldappc	A completely new Ldappc, Ldappc-ng, is included in the Grouper Toolkit in addition to the original Ldappc. Ldappc-ng uses SPML for provisioning and builds on Shibboleth's attribute resolver technology. The new attribute framework and permissions can be provisioned to LDAP with Ldappc-ng.
ESB integration	Grouper can provision group and folder information to and read in changes from an ESB using HTTPS or XMPP.
Subject picker	External applications can use the Grouper subject picker UI component to pick people.
Lite UI skinning	Lite UI can look like application that launches it, and can have customized text and instructions.
Flattened memberships	A new, additional flattened memberships table improves notification performance and lays the foundation for further read performance enhancement through the Grouper Toolkit.

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

Improvements & Fixes

UI sorting and paging of memberships	The effective membership list was improved to sort and page correctly
Composite groups and memberships	Ldappc had trouble provisioning include/exclude groups
Grouper UI access control lists	Grouper UI ACL's were straightened out
Ldappc-ng is its own project	Separates lots of jars from main Grouper API
XML Import/Export improvements	XML import/export uses less memory, is faster, and 100% of the Groups Registry is exported and imported.

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 to Grouper v1.6.2 from Grouper v1.6.1

You should just need to use the latest grouper.jar in the module you are using (WS, loader, etc), replace the existing grouper.jar, and restart the service. For the UI if you want GRP-496, you will need to rebuild the UI.

Upgrading from Grouper v1.5.*

- You should get the latest v1.6 (e.g. v1.6.2) 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.
- Once you prevent users from making updates to your Grouper instance, run the changeLogTempToChangeLog daemon to clear out the temp changelog using the v1.5 API. Here's an example using GSH.

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

- Before performing any upgrade steps, export your Grouper registry. Options include performing a database backup or using the xmlexport utility in Grouper. However this utility in v1.5 of the API does not support exporting and importing of the attribute framework introduced in v1.5, including permission management related objects. You can run the utility like this: ./bin/gsh.sh -xmlexport GrouperSystem backup.xml
- Using the 1.6.0 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..

```
user-152-3-214-219:grouper shilen$ ./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: 1.6.0, build date: 2010/06/04 15:06:22, 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: sa@jdbc:hsqldb:hsql://localhost/grouper
sources.xml read from: /srv/grouper/conf/sources.xml
sources.xml groupersource id: g:gsa
sources.xml jdbc source id: jdbc: GrouperJdbcConnectionProvider
This db user 'sa' and url 'jdbc:hsqldb:hsql://localhost/grouper' are allowed to be changed in the
grouper.properties
Continuing...
Grouper ddl object type 'Grouper' has dbVersion: 22 and java version: 23
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 f
or each env),
script file is:
/srv/grouper/ddlScripts/grouperDdl_20100604_15_41_14_838.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_20100604_15_41_14_838.sql
```

- In this example above, an SQL script called /srv/grouper/ddlScripts/grouperDdl_20100604_15_41_14_838.sql was created.
- Review the script to make sure it looks okay. The script will be backing up, dropping, and recreating the table GROUPER_ATTRIBUTE_ASSIGN_VALUE. It will also drop and recreate views, some constraints, and some indexes.
    - 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.
    - If using postgres or hsql and 1.6.0, you might want to manually drop all your grouper views, since some depend on others, and the script wont do it
    - If using postgres or hsql and 1.6.1+, 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.
    - If you are using mysql with certain international character sets, then change:
      CREATE INDEX attribute_field_value_idx ON grouper_attributes (field_id, value);
      To
      CREATE INDEX attribute_field_value_idx ON grouper_attributes (field_id, value(960));
- 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..

```
user-152-3-214-219:grouper shilen$ ./bin/gsh.sh -registry -runsqlfile ddlScripts
/grouperDdl_20100604_15_41_14_838.sql
Using GROUPER_HOME: /srv/grouper
Using GROUPER_CONF: /srv/grouper/conf
Using JAVA: java
using MEMORY: 64m-512m
This db user 'sa' and url 'jdbc:hsqldb:hsql://localhost/grouper' are allowed to be changed in the
grouper.properties
Continuing...
Script was executed successfully

Grouper starting up: version: 1.6.0, build date: 2010/06/04 15:06:22, 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: sa@jdbc:hsqldb:hsql://localhost/grouper
sources.xml read from: /srv/grouper/conf/sources.xml
sources.xml groupersource id: g:gsa
sources.xml jdbc source id: jdbc: GrouperJdbcConnectionProvider
```

- At this point, your DDL has been upgraded to v1.6.0. It should be safe to enable access to your Grouper instance now (UI, WS, etc). However, you should complete this next step before starting the Grouper Loader. Starting with v1.6.0, we're now populating a flat memberships table (along with other new dependent tables) to prevent duplicate notifications if a subject is a member of a group in multiple ways.
    - First be sure you have merged your copy of grouper-loader.properties with the v1.6.0 copy and make sure that the changeLogTempToChangeLog daemon is enabled.
    - Analyze your tables. At minimum, be sure to analyze grouper_group_set, grouper_memberships, grouper_groups, and grouper_stems.
    - And finally, populate the flat tables. Here's an example using GSH.

```
gsh 0% new SyncFlatTables().sendNotifications(false).syncAllFlatTables();


Searching for the number of missing flat groups
Found 0 missing flat groups


Searching for the number of missing flat stems
Found 0 missing flat stems


Searching for the number of missing flat attribute defs
Found 0 missing flat attribute defs


Searching for the number of missing flat memberships
Found approximately 7 missing flat memberships. The count is an estimate because it may include duplicates.
Retrieving batch of up to 100000 memberships
Done making 7 of approximately 7 updates


Searching for bad flat memberships
Found 0 bad flat memberships


Searching for bad flat groups
Found 0 bad flat groups


Searching for bad flat stems
Found 0 bad flat stems


Searching for bad flat attribute defs
Found 0 bad flat attribute defs
java.lang.Long: 7
```

- If the flat table sync appears to hang after starting to populate flat memberships, you may need to re-analyze your tables. This time, include grouper_flat_memberships as well.
- If you haven't in the past, you should probably enable the Grouper Report to run daily. The changeLogTempToChangeLog daemon will update the flat tables, but among other things, the Grouper Report this will sync the flat tables if they get out of sync.
- Start the Grouper Loader.