v2.3 Upgrade Instructions from v2.2

Upgrading to Grouper 2.3 from Grouper 2.2

Using the Grouper Upgrader can simplify your upgrade process. Here is a movie demonstrating the Grouper upgrader. The upgrader can upgrade an installed env of the API, UI, WS, client, PSP, etc. If you dont have a build script to manage multiple envs, you might want to use the upgrader.

Important Changes in Grouper 2.3 that impact the upgrade

Inherited Privileges: The Grouper v2.3 UI has support for privilege inheritance. By default, if you are an admin on a folder, you can assign inherited privileges on it. Note that one potential side effect of this feature is that it allows end users to gain access to sub-folders and groups because they have admin access to a parent folder. In most cases, this is expected behavior because folders are typically delegated and managed hierarchically. However, if you do not allow parent folder admins to have access to all child objects, then you may want to disable this feature. You have the option to lock this feature down so only Grouper admins can use it or people in a certain group.

Other items before upgrading

1. You may want to have your DBAs make sure you are not close to running out of tablespace. In general, it may be useful to have your DBAs available when you upgrade.
2. If you have views that other systems use, you could replace them as tables before beginning.
3. If you have other systems using Grouper, you could temporarily disable them.

Upgrade Steps

1. You should get v2.3 versions of the Grouper API, Grouper UI, Grouper WS, Grouper Daemon, etc. from the Grouper Downloads page. You will need to merge configuration files and JARs.
2. Stop the Grouper Daemon. Once you prevent users from making updates to your Grouper instance, run the changeLogTempToChangeLog daemon to clear out the temp changelog using your existing v2.2 API. Here's an example using GSH.

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

3. Before performing any upgrade steps, export your Grouper registry. Options include performing a database backup (recommended) or using the XML Export utility in Grouper (not recommended since certain features may not get exported).
4. Using the 2.3 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   Note you may need to increase memory. For instance..

```
$ export MEM_MAX=2000m
$ ./bin/gsh.sh -registry -check
Using GROUPER_HOME: /opt/grouper
Using GROUPER_CONF: /opt/grouper/conf
Using JAVA: java
using MEMORY: 64m-2000m
Grouper starting up: version: 2.3.0, build date: 2016/04/20 16:15:04, env: <no label configured>
grouper.properties read from: /opt/grouper/conf/grouper.properties
Grouper current directory is: /opt/grouper
log4j.properties read from:   /opt/grouper/conf/log4j.properties
Grouper is logging to file:   /opt/grouper/logs/grouper_debug.log, /opt/grouper/logs/grouper_error.log,
at min level INFO for package: edu.internet2.middleware.grouper, based on log4j.properties
grouper.hibernate.properties: /opt/grouper/conf/grouper.hibernate.properties
grouper.hibernate.properties: sa@jdbc:hsqldb:hsql://localhost:9001/grouper
sources.xml read from:        /opt/grouper/conf/sources.xml
sources.xml groupersource id: g:gsa
sources.xml groupersource id: grouperEntities
sources.xml jdbc source id:   jdbc: GrouperJdbcConnectionProvider
This db user 'sa' and url 'jdbc:hsqldb:hsql://localhost:9001/grouper' are allowed to be changed in the
grouper.properties
Continuing...
Grouper ddl object type 'Grouper' has dbVersion: 29 and java version: 30
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:
/opt/grouper/ddlScripts/grouperDdl_20160420_16_15_53_708.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 /opt/grouper/ddlScripts/grouperDdl_20160420_16_15_53_708.sql
```

    a. In this example above, an SQL script called /opt/grouper/ddlScripts/grouperDdl_20160420_16_15_53_708.sql was created.
    b. Postgres only - 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 don't see foreign keys being dropped at the top of the script, manually drop all foreign keys before running the script.
    c. Postgres and hsql only - 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.

5. Run the SQL script.

If you are running via GSH, make sure this is in log4j.properties so that you know which line of the script is currently executing to see progress and troubleshoot

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

To do this, run: gsh -registry -runsqlfile /path/to/sql/file.sql For instance..

```
$ ./bin/gsh.sh -registry -runsqlfile /opt/grouper/ddlScripts/grouperDdl_20160420_16_15_53_708.sql
Using GROUPER_HOME: /opt/grouper
Using GROUPER_CONF: /opt/grouper/conf
Using JAVA: java
using MEMORY: 64m-2000m
This db user 'sa' and url 'jdbc:hsqldb:hsql://localhost:9001/grouper' are allowed to be changed in the
grouper.properties
Continuing...
Script was executed successfully

Grouper starting up: version: 2.3.0, build date: 2016/04/20 16:15:04, env: <no label configured>
grouper.properties read from: /opt/grouper/conf/grouper.properties
Grouper current directory is: /opt/grouper
log4j.properties read from:   /opt/grouper/conf/log4j.properties
Grouper is logging to file:   /opt/grouper/logs/grouper_debug.log, /opt/grouper/logs/grouper_error.log,
at min level INFO for package: edu.internet2.middleware.grouper, based on log4j.properties
grouper.hibernate.properties: /opt/grouper/conf/grouper.hibernate.properties
grouper.hibernate.properties: sa@jdbc:hsqldb:hsql://localhost:9001/grouper
sources.xml read from:        /opt/grouper/conf/sources.xml
sources.xml groupersource id: g:gsa
sources.xml groupersource id: grouperEntities
sources.xml jdbc source id:   jdbc: GrouperJdbcConnectionProvider
Grouper note: auto-created stem: etc:attribute:messages
Grouper note: auto-created role: etc:attribute:messages:grouperMessageRole
Grouper note: auto-created attributeDef: etc:attribute:messages:grouperMessageTopicDef
Grouper note: auto-created attributeDef: etc:attribute:messages:grouperMessageQueueDef
Grouper note: auto-created stem: etc:attribute:messages:grouperMessageTopics
Grouper note: auto-created stem: etc:attribute:messages:grouperMessageQueues
```

Note that if one of the SQL statements in the script fails, the process will abort leaving the rest of the SQL statements from executing. If this happens, in most cases, you can't just re-run the full script since re-executing some of the DDL changes that previously succeeded would fail now (e.g. dropping a view or constraint that was previously dropped successfully.) You could edit the script to remove the statements that previously succeeded in order to re-execute the statement that failed and the ones after it. Or you can run the previous step again to generate a new SQL script.

6. Now that the DDL updates have been made, there is an additional GSH command that needs to be run. To do this, run: gsh ../misc /postGrouper2_3_0Upgrade.gsh (The gsh script is in the "misc" directory.) Note you should check the output to make sure no errors are thrown. If you see an error, it is safe to re-run. For instance..

```
$ ./bin/gsh.sh misc/postGrouper2_3_0Upgrade.gsh
Using GROUPER_HOME: /opt/grouper
Using GROUPER_CONF: /opt/grouper/conf
Using JAVA: java
using MEMORY: 64m-2000m
Grouper starting up: version: 2.3.0, build date: 2016/04/20 16:15:04, env: <no label configured>
grouper.properties read from: /opt/grouper/conf/grouper.properties
Grouper current directory is: /opt/grouper
log4j.properties read from:   /opt/grouper/conf/log4j.properties
Grouper is logging to file:   /opt/grouper/logs/grouper_debug.log, /opt/grouper/logs/grouper_error.log,
at min level INFO for package: edu.internet2.middleware.grouper, based on log4j.properties
grouper.hibernate.properties: /opt/grouper/conf/grouper.hibernate.properties
grouper.hibernate.properties: sa@jdbc:hsqldb:hsql://localhost:9001/grouper
sources.xml read from:        /opt/grouper/conf/sources.xml
sources.xml groupersource id: g:gsa
sources.xml groupersource id: grouperEntities
sources.xml jdbc source id:   jdbc: GrouperJdbcConnectionProvider
Type help() for instructions
Error: Cannot properly read UTF string from resource: grouperUtf8.txt: ''


##########################################
# Grouper 2.3.0 Upgrade Step 1/1: Remove grouperLoaderLdapErrorUnresolvable attribute
##########################################
edu.internet2.middleware.grouper.GrouperSession:
5868a5370afd4941bf3f340bf632546f,'GrouperSystem','application'
edu.internet2.middleware.grouper.attr.AttributeDefName: AttributeDefName[name=etc:attribute:loaderLdap:
grouperLoaderLdapErrorUnresolvable,uuid=799596896dd0426fb4c4e8edf9bd8a98]
Successfully removed attribute.
```

7. Analyze your tables. (To avoid any performance issues later.)
8. Start the Grouper Daemon and all other Grouper components (UI/WS).
9. The Grouper member table (grouper_members) now has a new column to store subject identifiers. Post 2.3.0, this will be used to help improve Grouper's performance in various aspects. You will need to configure your sources.xml file and sync the new column. For details: Subject Identifier column in member table
10. By default any folder owner can assign inherited privileges, which means they can get control of any descendant object. If you do not want this you can lock down control

See Also

Release Notes for Grouper 2.3