

Upgrade notes from Grouper v1.6 to v1.7

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

Upgrade notes from Grouper v1.6 to v1.7 (similar for v2.0)

This upgrade was pretty painless, it took a while (~5 hour outage including testing), and there was little or no outage during the upgrade. Note this is for an oracle upgrade, YMMV...

Here are the steps that were followed:

- move drop views to end of SQL upgrade script (since we have systems that depend on the views)
- disable nagios on fastGrouperProdDaemon
- select count of memberships of important groups, so we can sanity test the deployment (e.g. employees, some orgs, courses, etc)
- backup membership lite view to a table
- turn off daemon, ldap sync
- readonly ws/ui (set this in the grouper.properties and bounce)
- backup db (DBA)
- increase tablespace for schema
- backup old webapp dir

#####

clean out old entries from the change log:

```
create table change_log_entry_temp as select * from grouper_change_log_entry gcle where gcle.SEQUENCE_NUMBER > 2500000;
truncate table grouper_change_log_entry;
insert into grouper_change_log_entry (select * from change_log_entry_temp);
commit;
analyze table grouper_change_log_entry compute statistics;
drop table change_log_entry_temp;
```

#####

run rule on root stem and secure space stem:

```
grouperSession = GrouperSession.startRootSession();
rootStem = StemFinder.findRootStem(grouperSession);
allowedStem = StemFinder.findByName(grouperSession, "penn:isc:ait:apps:secureSpace:spaces", true);
allowedStemTest = StemFinder.findByName(grouperSession, "test:isc:ait:apps:secureSpace:spaces", true);
```

```
RuleApi.vetoSubjectAssignInFolderIfNotInGroup(SubjectFinder.findRootSubject(), rootStem, null, false, "grouperExternal", Stem.Scope.SUB, "rule.entity.
cannot.be.external", "Person cannot be assigned if an external user");
```

```
RuleApi.vetoSubjectAssignInFolderIfNotInGroup(SubjectFinder.findRootSubject(), allowedStem, null, true, "grouperExternal", Stem.Scope.SUB, "rule.
entity.can.be.external", "Person can be external");
```

```
RuleApi.vetoSubjectAssignInFolderIfNotInGroup(SubjectFinder.findRootSubject(), allowedStemTest, null, true, "grouperExternal", Stem.Scope.SUB,
"rule.entity.can.be.external", "Person can be external");
```

#####

```
select count(1) from grouper_memberships_lw_v
```

```
2500566
2501357
```

```
select count(1) from grouper_groups
```

```
69239
69240
```

```
select count(1) from grouper_stems
```

```
7171
7174
```

```
select from shibboleth and ldap feeder tables
```

```
select count(1) from grouper_memberships_lw_v where group_name = 'penn:community:employee:org:TOPU:UNIV:UADM:91XX:AIS:9147:
9147_personorg'
```

```
23
```

#####

readonly mode:

```
/opt/appserv/tomcat_2v/webapps/grouper/WEB-INF/classes/grouper.hibernate.properties
/opt/appserv/tomcat_3b/webapps/grouperWs/WEB-INF/classes/grouper.hibernate.properties
```

#####

```
select 'ANALYZE TABLE ' || table_name || ' COMPUTE STATISTICS;' as script from user_Tables where table_name like 'GROUPER%'
```

```
ANALYZE TABLE GROUPERORGS_HIERARCHICAL COMPUTE STATISTICS;
ANALYZE TABLE GROUPERORGS_POC_ORGS COMPUTE STATISTICS;
ANALYZE TABLE GROUPERORGS_POC_ORG_ASSIGN COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_ATTRIBUTES COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_ATTRIBUTE_ASSIGN COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_ATTRIBUTE_ASSIGN_VALUE COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_ATTRIBUTE_DEF COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_ATTRIBUTE_DEF_NAME COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_ATTRIBUTE_DEF_NAME_SET COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_ATTRIBUTE_DEF_SCOPE COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_ATTR_ASSIGN_ACTION COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_ATTR_ASSIGN_ACTION_SET COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_AUDIT_ENTRY COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_AUDIT_TYPE COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_CHANGE_LOG_CONSUMER COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_CHANGE_LOG_ENTRY COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_CHANGE_LOG_ENTRY_TEMP COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_CHANGE_LOG_TYPE COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_COMPOSITES COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_DDL COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_FIELDS COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_FLAT_ATTRIBUTE_DEF COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_FLAT_GROUPS COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_FLAT_MEMBERSHIPS COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_FLAT_STEMS COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_GROUPS COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_GROUPS_TYPES COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_GROUP_SET COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_LOADER_LOG COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_MEMBERS COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_MEMBERSHIPS COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_ROLE_SET COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_STEMS COMPUTE STATISTICS;
ANALYZE TABLE GROUPER_TYPES COMPUTE STATISTICS;
```

- run the ddl

```
[appadmin@johns bin]$ dos2unix gsh.sh
dos2unix: converting file gsh.sh to UNIX format ...
[appadmin@johns bin]$ chmod +x gsh.sh
```

```
[appadmin@johns bin]$ ./gsh.sh -registry
```

Get script and run in toad after making indexes unusable

make indexes unusable before the inserts:

```
select * from user_indexes;
cur_idx usr_idx% ROWTYPE;
v_sql VARCHAR2(1024);

BEGIN
OPEN usr_idx;
LOOP
FETCH usr_idx INTO cur_idx;
EXIT WHEN NOT usr_idx%FOUND;

v_sql:= 'ALTER INDEX ' || cur_idx.index_name || ' UNUSABLE';
END LOOP;

END;
```

Compile invalid views in toad, reenale indexes

Add EPPN to the SQL person source table and to the sources.xml

```
#####
compile all invalid views
```

- get counts from DB, compare to original number

- test atlassian (document)
- test eforms (document)
- test secureshare
- post new grouper client to upenn.edu
- download and test the client
- test external subjects (document)
- run a deep registry gsh check and see if it is ok
- analyze the rest of the tables, starting with grouper_group_set
- enable nagios, ldap sync
- send email to users saying it is available
- check daemons the day after and the day after to make sure ok