DDL in Grouper v2.5+

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

DDL is the database table and view structure.

DDL upgrade

Method	How	Description

Auto	Set auto ddl version to whatever version you are on. In the container you can use the env var:
	`GROUPER_AUTO_DDL_UPTOVERSION=2.6.*`
	Or you can set this in the configuration:
	grouper.hibernate.properties (for 2.5)
	`registry.auto.ddl.upToVersion = 2.5.*`
	grouper.hibernate.properties (for 2.6)
	`registry.auto.ddl.upToVersion = 2.6.*`
		Just start the Grouper UI or WS or GSH or the daemon, and Grouper will auto-upgrade the DDL

Manual run script	`docker exec -u tomcat -it grouper-ui /bin/bash`
	`cd /opt/grouper/grouperWebapp/WEB-INF/bin`
	`./gsh.sh -registry -check -runscript`
		This will generate a script and run it in your database

Manual generate script	`docker exec -u tomcat -it grouper-ui /bin/bash`
	`cd /opt/grouper/grouperWebapp/WEB-INF/bin`
	`./gsh.sh -registry -check`
		This will generate a script that you can run against your database

If a DDL script does not complete successfully, you will need to trace through the script and the database to see where it stopped and what needs to be run. This is not as difficult as it seems. e.g.

- v2.5 example: here is a postgres upgrade script for 2.5.39
- v2.6 scripts: https://github.com/Internet2/grouper/tree/GROUPER_2_6_BRANCH/grouper/conf/ddl

If you see an error in the logs when running DDL, you might be able to ignore it since the upgrade might have worked. e.g.

```
ERROR SqlExceptionHelper.logExceptions(142) - [] - Unknown column 'member0_.subject_identifier1' in 'field list'
```

Database upgrades in v2.5

Verify the DDL version with this query:

```
select * from grouper_ddl where object_name = 'Grouper'
```

Version	Description	Upgrade type	DDL version	Verify
v2.5.22	Add password and sync tables, adjust some views		32	This should not have an error `select count(1) from grouper_sync`

v2.5.31	Add recent membership tables and views, add PIT views, add grouper member columns for USDU		33	This should not have an error
				`select distinct subject_resolution_resolvable from grouper_members;`
				`select count(1) from grouper_recent_mships_load_v;`

v2.5.34	Add config PIT table and grouper_file table		34	This should not have an error `select count(1) from grouper_file;`
v2.5.38	Add provisioning log and sync start columns		35	This should not have an error `select count(1) from grouper_sync_membership_v;`
v2.5.40	Add provisioning error codes, and provisioning membership view		36	This should not have an error `select error_code from grouper_sync_membership;`
v2.5.51	Add metadata column to grouper_sync_group		37	This should not have an error `select count(*) from grouper_sync_group where metadata_json is null;`
v2.6.1	Add zoom user table, adjust grouper_password columns		38	This should not have an error `select count(1) from grouper_prov_zoom_user;`
v2.6.5	Adjust zoom status col, add table for failsafes, add col for usdu_eligible		39	This should not have an error `select count(1) from grouper_members where subject_resolution_eligible is null;`

v2.6.6	Add subject identifier and email cols to grouper_members, add metadata_json to grouper_sync_members	Minimal	40	This should not have an error `select count(1) from grouper_members where subject_identifier1 is null;`
v2.6.8	Create table grouper_prov_duo_user	Minimal	41	This should not have an error `select count(1) from grouper_prov_duo_user;`
v2.6.14	Add columns job_message_clob and job_message_bytes to grouper_loader_log	Minimal	42	This should not have an error `select * from grouper_loader_log where job_message_clob is null and job_message_bytes is null;`

v2.6.16	Add table grouper_mship_req_change, add column grouper_members.id_index
	Remove foreign keys on grouper_stem_view_privilege
		Significant
		Stop updates when running
			43	These should not have an error
				`select count(1) from grouper_mship_req_change;`
				`select count(1) from grouper_members where id_index is null;`

v2.6.18	Update columns grouper_attribute_assign.disallowed and grouper_pit_attribute_assign.disallowed to: DEFAULT 'F' NOT NULL	Minimal	44	

v5.0.3	Add grouper_members internal_id column
	Add grouper_dictionary table
	Add grouper_data_provider table
	Add grouper_data_field table
	Add grouper_data_row table
	Add grouper_data_alias table
	Add grouper_data_field_assign table
	Add grouper_data_row_assign table
	Add grouper_data_row_field_assign table
	Add grouper_data_global_assign table
	Add grouper_data_field_assign_v view
	Add grouper_data_row_assign_v view
	Add grouper_data_row_field_asgn_v view
		Significant
		Stop updates when running
			45	This should not have an error
				`select count(1) from grouper_members where internal_id is null;`
				`select count(1) from grouper_dictionary;`
				`select count(1) from grouper_data_provider;`
				`select count(1) from grouper_data_field;`
				`select count(1) from grouper_data_row;`
				`select count(1) from grouper_data_alias;`
				`select count(1) from grouper_data_field_assign;`
				`select count(1) from grouper_data_row_assign;`
				`select count(1) from grouper_data_row_field_assign;`
				`select count(1) from grouper_data_global_assign;`
				`select count(1) from grouper_data_field_assign_v;`
				`select count(1) from grouper_data_row_assign_v;`
				`select count(1) from grouper_data_row_field_asgn_v;`

v5.0.4	Add grouper_fields internal_id column
	Add grouper_sql_cache_group table
	Add grouper_sql_cache_mship table
	Add grouper_sql_cache_mship_hst table
	Add grouper_sql_cache_group_v view
	Add grouper_sql_cache_mship_v view
	Add grouper_sql_cache_mship_hst_v view
		Significant
		Stop updates when running
			46	This should not have an error
				`select count(1) from grouper_fields where internal_id is null;`
				`select count(1) from grouper_sql_cache_group;`
				`select count(1) from grouper_sql_cache_mship;`
				`select count(1) from grouper_sql_cache_mship_hst;`
				`select count(1) from grouper_sql_cache_group_v;`
				`select count(1) from grouper_sql_cache_mship_v;`
				`select count(1) from grouper_sql_cache_mship_hst_v;`

Description

Grouper 2.5+ can handle DDL more efficiently and automatically. DDL can now also change during Grouper build number (previously we kept DDL largely to minor upgrades).

If Grouper is running the DDL automatically, or you run it from gsh manaually, or you run the script in your DB UI tool or whatever, if it fails part-way through, you need to grab the rest of the DDL scripts (from WEB-INF/ddlScripts) and run the rest manually. Grouper will not be able to start where it left off and you need to fix it.

The DDL that Grouper runs will be minimal for the task at hand and not drop and create all views or foreign keys.

On startup Grouper will see if the database is up to date (i.e. Java is a higher DDL version than the database), and if so a static DDL SQL script will either be run automatically or you can run the scripts manually or through GSH. If you are running manually, just start Grouper up (either GSH/daemon/UI/WS /scim/whatever, and it will log which DDL to use. These are the static scripts. There are scripts:

Note: <db> is either postgres (recommended), mysql, oracle, or hsql

Script	Description
GrouperDdl_Grouper_createDdlWorker_<db>.sql	Create the new (for 2.5) grouper_ddl_worker table which synchronizes which JVM does the DDL updates when multiple start up at once
GrouperDdl_Grouper_install_<db>.sql	Grouper DDL if the DB starts from scratch with no tables
GrouperDdl_Subject_install_<db>.sql	Subject tables (for the sample subject source, maybe not needed but thats ok)
GrouperDdl_Grouper_30_upgradeTo_31_<db>.sql	Upgrade script from v2.3 to v2.4
GrouperDdl_Grouper_31_upgradeTo_32_<db>.sql	Upgrade script from v2.4 to v2.5

Note: if upgrading from Grouper prior to v2.3, you need to upgrade to 2.3 first, then startup a 2.5 container to do the rest.

Check your DDL

If you are upgrading your DDL, you can run this:

```
./gsh.sh -registry -check
```

If your container is the same version as your database, you can run (from /opt/grouper/grouperWebapp/WEB-INF/bin

```
./gsh.sh -registry -check -deep
```

Note, the script generated by "-deep" **is not the one you should run**, but you will get a report that will help you figure out issues. Use this report and the full scripts above to try to fix things if your DDL is not correct.

You should see something like this:

```
SUCCESS: Database DDL is correct!

Note: Database version for Grouper: 32 (2.5.0)
Note: Java version for Grouper: 32 (2.5.0)
Success: Database version is the same as the Java codebase Grouper version
Success: Table 'grouper_attr_assign_action': Table is up to date.  7 columns, 1 indexes, 1 foreign keys.
Success: Table 'grouper_attr_assign_action_set': Table is up to date.  10 columns, 3 indexes, 3 foreign keys.
Success: Table 'grouper_attribute_assign': Table is up to date.  20 columns, 8 indexes, 8 foreign keys.
Success: Table 'grouper_attribute_assign_value': Table is up to date.  10 columns, 4 indexes, 1 foreign keys.
Success: Table 'grouper_attribute_def': Table is up to date.  28 columns, 3 indexes, 1 foreign keys.
Success: Table 'grouper_attribute_def_name': Table is up to date.  13 columns, 2 indexes, 2 foreign keys.
Success: Table 'grouper_attribute_def_name_set': Table is up to date.  10 columns, 3 indexes, 3 foreign keys.
Success: Table 'grouper_attribute_def_scope': Table is up to date.  9 columns, 1 indexes, 1 foreign keys.
Success: Table 'grouper_audit_entry': Table is up to date.  30 columns, 13 indexes, 1 foreign keys.
Success: Table 'grouper_audit_type': Table is up to date.  20 columns, 1 indexes, 0 foreign keys.
Success: Table 'grouper_change_log_consumer': Table is up to date.  6 columns, 1 indexes, 0 foreign keys.
Success: Table 'grouper_change_log_entry': Table is up to date.  16 columns, 15 indexes, 1 foreign keys.
Success: Table 'grouper_change_log_entry_temp': Table is up to date.  16 columns, 13 indexes, 0 foreign keys.
Success: Table 'grouper_change_log_type': Table is up to date.  19 columns, 1 indexes, 0 foreign keys.
Success: Table 'grouper_composites': Table is up to date.  9 columns, 8 indexes, 4 foreign keys.
Success: Table 'grouper_config': Table is up to date.  11 columns, 4 indexes, 0 foreign keys.
Success: Table 'grouper_ddl': Table is up to date.  5 columns, 1 indexes, 0 foreign keys.
Success: Table 'grouper_ddl_worker': Table is up to date.  5 columns, 1 indexes, 0 foreign keys.
Success: Table 'grouper_ext_subj': Table is up to date.  16 columns, 2 indexes, 0 foreign keys.
Success: Table 'grouper_ext_subj_attr': Table is up to date.  10 columns, 3 indexes, 1 foreign keys.
Success: Table 'grouper_fields': Table is up to date.  7 columns, 3 indexes, 0 foreign keys.
Success: Table 'grouper_group_set': Table is up to date.  22 columns, 15 indexes, 10 foreign keys.
Success: Table 'grouper_groups': Table is up to date.  21 columns, 17 indexes, 3 foreign keys.
Success: Table 'grouper_loader_log': Table is up to date.  27 columns, 2 indexes, 0 foreign keys.
Success: Table 'grouper_members': Table is up to date.  19 columns, 13 indexes, 0 foreign keys.
Success: Table 'grouper_memberships': Table is up to date.  16 columns, 21 indexes, 7 foreign keys.
Success: Table 'grouper_message': Table is up to date.  11 columns, 6 indexes, 1 foreign keys.
Success: Table 'grouper_password': Table is up to date.  16 columns, 1 indexes, 0 foreign keys.
Success: Table 'grouper_password_recently_used': Table is up to date.  4 columns, 0 indexes, 1 foreign keys.
Success: Table 'grouper_pit_attr_assn_actn': Table is up to date.  9 columns, 4 indexes, 1 foreign keys.
Success: Table 'grouper_pit_attr_assn_actn_set': Table is up to date.  11 columns, 6 indexes, 3 foreign keys.
Success: Table 'grouper_pit_attr_assn_value': Table is up to date.  12 columns, 8 indexes, 1 foreign keys.
Success: Table 'grouper_pit_attr_def_name': Table is up to date.  10 columns, 6 indexes, 2 foreign keys.
Success: Table 'grouper_pit_attr_def_name_set': Table is up to date.  11 columns, 6 indexes, 3 foreign keys.
Success: Table 'grouper_pit_attribute_assign': Table is up to date.  17 columns, 12 indexes, 8 foreign keys.
Success: Table 'grouper_pit_attribute_def': Table is up to date.  10 columns, 7 indexes, 1 foreign keys.
Success: Table 'grouper_pit_fields': Table is up to date.  9 columns, 5 indexes, 0 foreign keys.
Success: Table 'grouper_pit_group_set': Table is up to date.  19 columns, 18 indexes, 9 foreign keys.
Success: Table 'grouper_pit_groups': Table is up to date.  9 columns, 6 indexes, 1 foreign keys.
Success: Table 'grouper_pit_members': Table is up to date.  11 columns, 6 indexes, 0 foreign keys.
Success: Table 'grouper_pit_memberships': Table is up to date.  13 columns, 11 indexes, 5 foreign keys.
Success: Table 'grouper_pit_role_set': Table is up to date.  11 columns, 6 indexes, 3 foreign keys.
Success: Table 'grouper_pit_stems': Table is up to date.  9 columns, 6 indexes, 1 foreign keys.
Success: Table 'grouper_qz_blob_triggers': Table is up to date.  4 columns, 0 indexes, 1 foreign keys.
Success: Table 'grouper_qz_calendars': Table is up to date.  3 columns, 0 indexes, 0 foreign keys.
Success: Table 'grouper_qz_cron_triggers': Table is up to date.  5 columns, 0 indexes, 1 foreign keys.
Success: Table 'grouper_qz_fired_triggers': Table is up to date.  13 columns, 6 indexes, 0 foreign keys.
Success: Table 'grouper_qz_job_details': Table is up to date.  10 columns, 2 indexes, 0 foreign keys.
Success: Table 'grouper_qz_locks': Table is up to date.  2 columns, 0 indexes, 0 foreign keys.
Success: Table 'grouper_qz_paused_trigger_grps': Table is up to date.  2 columns, 0 indexes, 0 foreign keys.
Success: Table 'grouper_qz_scheduler_state': Table is up to date.  4 columns, 0 indexes, 0 foreign keys.
Success: Table 'grouper_qz_simple_triggers': Table is up to date.  6 columns, 0 indexes, 1 foreign keys.
Success: Table 'grouper_qz_simprop_triggers': Table is up to date.  14 columns, 0 indexes, 1 foreign keys.
Success: Table 'grouper_qz_triggers': Table is up to date.  16 columns, 12 indexes, 1 foreign keys.
Success: Table 'grouper_role_set': Table is up to date.  10 columns, 3 indexes, 3 foreign keys.
Success: Table 'grouper_stem_set': Table is up to date.  10 columns, 3 indexes, 3 foreign keys.
Success: Table 'grouper_stems': Table is up to date.  16 columns, 13 indexes, 3 foreign keys.
Success: Table 'grouper_sync': Table is up to date.  12 columns, 2 indexes, 0 foreign keys.
Success: Table 'grouper_sync_group': Table is up to date.  23 columns, 8 indexes, 1 foreign keys.
Success: Table 'grouper_sync_job': Table is up to date.  13 columns, 1 indexes, 1 foreign keys.
Success: Table 'grouper_sync_log': Table is up to date.  11 columns, 2 indexes, 1 foreign keys.
Success: Table 'grouper_sync_member': Table is up to date.  24 columns, 9 indexes, 1 foreign keys.
Success: Table 'grouper_sync_membership': Table is up to date.  14 columns, 6 indexes, 3 foreign keys.
Success: Table 'grouper_table_index': Table is up to date.  6 columns, 1 indexes, 0 foreign keys.
Success: View 'grouper_attr_asn_asn_attrdef_v': View is up to date.  23 columns.
```

```
Success: View 'grouper_attr_asn_asn_efmship_v': View is up to date.  27 columns.
Success: View 'grouper_attr_asn_asn_group_v': View is up to date.  24 columns.
Success: View 'grouper_attr_asn_asn_member_v': View is up to date.  24 columns.
Success: View 'grouper_attr_asn_asn_mship_v': View is up to date.  28 columns.
Success: View 'grouper_attr_asn_asn_stem_v': View is up to date.  24 columns.
Success: View 'grouper_attr_asn_attrdef_v': View is up to date.  14 columns.
Success: View 'grouper_attr_asn_efmship_v': View is up to date.  20 columns.
Success: View 'grouper_attr_asn_group_v': View is up to date.  16 columns.
Success: View 'grouper_attr_asn_member_v': View is up to date.  16 columns.
Success: View 'grouper_attr_asn_mship_v': View is up to date.  20 columns.
Success: View 'grouper_attr_asn_stem_v': View is up to date.  15 columns.
Success: View 'grouper_attr_assn_action_set_v': View is up to date.  10 columns.
Success: View 'grouper_attr_def_name_set_v': View is up to date.  10 columns.
Success: View 'grouper_attr_def_priv_v': View is up to date.  12 columns.
Success: View 'grouper_audit_entry_v': View is up to date.  45 columns.
Success: View 'grouper_aval_asn_asn_attrdef_v': View is up to date.  28 columns.
Success: View 'grouper_aval_asn_asn_efmship_v': View is up to date.  32 columns.
Success: View 'grouper_aval_asn_asn_group_v': View is up to date.  29 columns.
Success: View 'grouper_aval_asn_asn_member_v': View is up to date.  29 columns.
Success: View 'grouper_aval_asn_asn_mship_v': View is up to date.  33 columns.
Success: View 'grouper_aval_asn_asn_stem_v': View is up to date.  29 columns.
Success: View 'grouper_aval_asn_attrdef_v': View is up to date.  19 columns.
Success: View 'grouper_aval_asn_efmship_v': View is up to date.  25 columns.
Success: View 'grouper_aval_asn_group_v': View is up to date.  21 columns.
Success: View 'grouper_aval_asn_member_v': View is up to date.  21 columns.
Success: View 'grouper_aval_asn_mship_v': View is up to date.  25 columns.
Success: View 'grouper_aval_asn_stem_v': View is up to date.  20 columns.
Success: View 'grouper_change_log_entry_v': View is up to date.  30 columns.
Success: View 'grouper_composites_v': View is up to date.  15 columns.
Success: View 'grouper_ext_subj_invite_v': View is up to date.  13 columns.
Success: View 'grouper_ext_subj_v': View is up to date.  7 columns.
Success: View 'grouper_groups_v': View is up to date.  24 columns.
Success: View 'grouper_memberships_all_v': View is up to date.  24 columns.
Success: View 'grouper_memberships_lw_v': View is up to date.  7 columns.
Success: View 'grouper_memberships_v': View is up to date.  23 columns.
Success: View 'grouper_mship_attrdef_lw_v': View is up to date.  6 columns.
Success: View 'grouper_mship_stem_lw_v': View is up to date.  6 columns.
Success: View 'grouper_perms_all_v': View is up to date.  27 columns.
Success: View 'grouper_perms_assigned_role_v': View is up to date.  20 columns.
Success: View 'grouper_perms_role_subject_v': View is up to date.  27 columns.
Success: View 'grouper_perms_role_v': View is up to date.  27 columns.
Success: View 'grouper_pit_attr_asn_value_v': View is up to date.  18 columns.
Success: View 'grouper_pit_memberships_all_v': View is up to date.  19 columns.
Success: View 'grouper_pit_perms_all_v': View is up to date.  47 columns.
Success: View 'grouper_pit_perms_role_subj_v': View is up to date.  47 columns.
Success: View 'grouper_pit_perms_role_v': View is up to date.  47 columns.
Success: View 'grouper_role_set_v': View is up to date.  10 columns.
Success: View 'grouper_roles_v': View is up to date.  23 columns.
Success: View 'grouper_rpt_composites_v': View is up to date.  2 columns.
Success: View 'grouper_rpt_group_field_v': View is up to date.  5 columns.
Success: View 'grouper_rpt_groups_v': View is up to date.  8 columns.
Success: View 'grouper_rpt_members_v': View is up to date.  4 columns.
Success: View 'grouper_rpt_roles_v': View is up to date.  7 columns.
Success: View 'grouper_rpt_stems_v': View is up to date.  10 columns.
Success: View 'grouper_rules_v': View is up to date.  30 columns.
Success: View 'grouper_service_role_v': View is up to date.  15 columns.
Success: View 'grouper_stem_set_v': View is up to date.  10 columns.
Success: View 'grouper_stems_v': View is up to date.  19 columns.
```

Configuration

You need to add an entry to your **grouper.hibernate.properties** to identify that the environment should auto-update DDL to any version in 2.5. This is recommended. This will not allow a 2.6 container to automatically change the DDL, but any container in 2.5 that is newer than the DB will update DDL and they are backwards compatible (if you dont update a WS container, but do update the UI, while still in 2.5.*, both will work). i.e. try not to drop columns. Note the database user that you connect as needs to be able to change DDL in the schema that it connects to. If you don't make this setting for whatever reason, you need to manually make DDL changes when new containers require it. We will identify if that is the case on the release notes page for a container. Note: if you don't want auto-DDL, you can change a setting so that Grouper will not log an error reminding you to change this setting. This is not set by default since we do not want to change DDL without you consciously configuring that you are expecting it.

```
# what version should we auto install DDL up to.  You should put the major and minor version here (e.g. 2.5.
*).  Or you could go to a build number if you like,
# or nothing to not auto DDL.  e.g. 2.5.32     or     2.5.*
# {valueType: "string"}
registry.auto.ddl.upToVersion = 2.5.*
```

Full settings: **grouper.hibernate.properties**

```
#######################################
## Initialization settings
#######################################

# what version should we auto install DDL up to.  You should put the major and minor version here (e.g. 2.5.
*).  Or you could go to a build number if you like,
# or nothing to not auto DDL.  e.g. 2.5.32     or     2.5.*
# {valueType: "string"}
registry.auto.ddl.upToVersion =

# if you are consciously not doing auto-ddl, set this to true
# {valueType: "boolean", required: true}
registry.auto.ddl.dontRemindMeAboutUpToVersion = false

# its ok if a WS is a little behind on DDL as the UI.  If the major and minor version are the same then its not
considered a mismatch
# {valueType: "boolean", required: true}
registry.auto.ddl.okIfSameMajorAndMinorVersion = true
```

Static scripts vs ddlutils

Note: if you are pre v2.3, you need to upgrade to v2.3 or v2.4 before upgrading to v2.5

Command	Strategy
startup	static SQL
gsh.sh -registry -check	static SQL
gsh.sh -registry -check -runscript	static SQL
gsh.sh -registry -check -drop	drop and then static SQL
gsh.sh -registry -check -useDdlUtils	ddlUtils
... -deep ...	ddlUtils

Grouper DDL locking

There is a new table grouper_ddl_worker which will be auto-created (if you are doing auto-DDL): grouper_ddl_worker that JVMs can lock on to do DDL or wait for another JVM to finish doing DDL. This is so multiple JVMs dont attempt this at once.

Grouper DDL versioning

Each DDL version is now correlated with a Grouper version. e.g.

From:

```
Grouper ddl object type 'Grouper' has dbVersion: 31 and java version: 32
```

To:

```
Grouper ddl object type 'Grouper' has dbVersion: 31 (2.4.0) and java version: 32 (2.5.0)
```

Grouper view updates

Database	Supports "replace view"	Replace maintains grants	Replace ok with views on views
hsql	no	NA	NA
mysql	yes	yes	yes
postgres	yes	yes	yes
oracle	yes	yes	yes

Some DDL/SQL to test with

```
create or replace view whatever as select * from grouper_groups where name like '%';

create view whatever2 as select * from whatever;

grant select on whatever to grouper_v2_5a;

select * from whatever2;

drop view whatever;

select * from whatever2;
```