v1.5 Release Notes

Unable to render {include} The included page could not be found.

Release Notes for Grouper v1.5

Grouper v1.5.3 contains 6 fixes and improvements, including a readonly mode for grouper

Grouper v1.5.2 contains 6 fixes and improvements, including support for MS SQL Server and minor enhancements to the Lite UI.

Grouper v1.5.1 fixes three bugs in the 1.5.0 release.

Grouper v1.5.0 includes 57 fixes and improvements over v1.4.2. See the full list.

New Features

Lite UI	An AJAX-based widget simplifies some end user interactions. Available in two ways: free-standing, or integrated within the Administrative UI.
Audit	Who took which management actions when is recorded and made available for viewing and reporting.
Move and Copy	Move or copy groups and folders to other folders, with the option to preserve old group names so that applications may continue to refer to the old name.
Notification	Real-time notification of group, folder, membership, and privilege changes are available through the Grouper API.
Attribute framework	Assign custom attributes to groups, memberships, folders, and other attributes.
Roles and Permissions	Support for Roles and Role hierarchies. Permissions can be attached to Roles or to Memberships in Roles.
Shibboleth integration	The integration of Shibboleth's Attribute Resolver within LDAPPC provides substantial attribute calculation capabilities. Also, memberships and group attributes can be accessed directly by a Shibboleth IdP.

Most of these implementations are in an initial form and will be completed as listed in the Grouper Product Roadmap.

Improvements & Fixes

Performance	A fundamental change to Grouper's underlying relational schema makes it far faster at write operations. Write time now is nearly independent of the number of indirect memberships involved.
Bad membership fix	Schema change that makes it impossible for spurious "bad memberships" to occur in direct or indirect memberships.
Membership enable and disable dates	Membership assignments can have enabled/disabled dates where the membership might be enabled in the future, or disabled after a certain period of time.
LDAPPC	Improved performance, configurability, and integration with Active Directory, as well as several other enhancements and fixes.
Administrative UI	Updated to enable AJAX support, enable clustering, and several other fixes and enhancements.

Many other fixes and more minor 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, and Subject API.

Upgrading from Grouper v1.4.2

- You should get v1.5.0 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.
- There have been a lot of updates to the Grouper schema for v1.5.0. We now have more database tables for the attribute framework, changelog, and audit. Also the default group attributes (name, display_name, extension, display_extension, description) are now stored in the grouper_groups table rather than the grouper_attributes table. And we've also refactored how we handle effective memberships. Effective memberships are no longer stored in the grouper_memberships table, but are rather formed by joining a Membership row with a GroupSet row in the database.
- Before performing any upgrade steps, using your current v1.4.2 API, export your Grouper registry to an XML file. For instance -- ./bin/gsh.sh -xmlexport GrouperSystem backup.xml
- There are two ways you can upgrade. Both options are described below.

Upgrade Method 1

- This option is generally easier, but also more time consuming for large deployments.
- Make sure you have an XML backup as described above. And make sure the XML file is complete and there were no errors generated in your logs during the export.
- Drop all of the Grouper objects from your database
- Using the 1.5.0 API, create the new database schema. To do this, run: gsh -registry -runscript For instance..

```
[root@idms-devel-01 grouper]# ./bin/gsh.sh -registry -runscript
Using GROUPER_HOME: /srv/grouper
Using GROUPER_CONF: /srv/grouper/conf
Using JAVA: java
using MEMORY: 64m-512m
Grouper starting up: version: 1.5.0, build date: 2009/11/14 13:14:33, env: <no label configured>
grouper.properties read from: /srv/grouper/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: groups@jdbc:oracle:thin:@imsdev-db.oit.duke.edu:1612:IMSDEV
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 or deny db urls and users in the grouper.properties)
Are you sure you want to schemaexport all tables (dropThenCreate=F,writeAndRunScript=T) in db user
'groups', db url 'jdbc:oracle:thin:@imsdev-db.oit.duke.edu:1612:IMSDEV'? (y|n):
y
Continuing...
Grouper ddl object type 'Grouper' has dbVersion: 0 and java version: 22
Grouper ddl object type 'Subject' has dbVersion: 0 and java version: 1
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/grouperDdl_20091114_14_34_58_674.sql
Script was executed successfully
```

- Use XML Import to import your Grouper registry using the backup you created. For instance..

```
./bin/gsh.sh -xmlimport GrouperSystem backup.xml
```

Upgrade Method 2

- Using the 1.5.0 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..

```
[root@idms-devel-01 grouper]# ./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: 1.5.0, build date: 2009/11/14 13:14:33, env: <no label configured>
grouper.properties read from: /srv/grouper/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: groups@jdbc:oracle:thin:@imsdev-db.oit.duke.edu:1612:IMSDEV
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 or 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
'groups', db url 'jdbc:oracle:thin:@imsdev-db.oit.duke.edu:1612:IMSDEV'? (y|n):
y
Continuing...
Grouper ddl object type 'Grouper' has dbVersion: 13 and java version: 22
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 - g
enerate a new one for each env),
script file is:
/srv/grouper/grouperDdl_20091114_13_16_01_466.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/grouperDdl_20091114_13_16_01_466.sql
```

- In this example above, an SQL script called /srv/grouper/grouperDdl_20091114_13_16_01_466.sql was created.
- Review the script to make sure it looks okay. The script shouldn't be dropping or truncating any tables. However, it will drop and recreate views, some constraints, and some indexes.
    - If using postgres, you might see some tables being backed up and recreated
    - 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 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..

```
[root@idms-devel-01 grouper]# ./bin/gsh.sh -registry -runsqlfile /srv/grouper
/grouperDdl_20091114_13_16_01_466.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 or deny db urls and users in the grouper.properties)
Are you sure you want to run the sql file in db user 'groups', db url 'jdbc:oracle:thin:@imsdev-db.oit.
duke.edu:1612:IMSDEV'? (y|n):
y
Continuing...
Script was executed successfully

Grouper starting up: version: 1.5.0, build date: 2009/11/14 13:14:33, env: <no label configured>
grouper.properties read from: /srv/grouper/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: groups@jdbc:oracle:thin:@imsdev-db.oit.duke.edu:1612:IMSDEV
sources.xml read from:        /srv/grouper/conf/sources.xml
sources.xml groupersource id: g:gsa
sources.xml jdbc source id:   jdbc: GrouperJdbcConnectionProvider
```

- If you script fails on:

```
CREATE UNIQUE INDEX membership_uniq_idx ON GROUPER_MEMBERSHIPS (OWNER_ID, MEMBER_ID, FIELD_ID);

then you need to find the rows which have dupes and delete ONE of them, e.g.

select count(*), OWNER_ID, MEMBER_ID, FIELD_ID, max(id) from grouper_memberships group by OWNER_ID, MEMBER_ID,
FIELD_ID having count(*) > 1;

This returned several rows (of count 2), so I just deleted these.

delete from grouper_memberships where (owner_id, member_id, field_id, id, 2) in
(select OWNER_ID, MEMBER_ID, FIELD_ID, max(id), count(*) from grouper_memberships
group by OWNER_ID, MEMBER_ID, FIELD_ID having count(*) > 1);
commit;
```

- If you are upgrading from 1.4, you might have rows with null hibernate_version_number, and hibernate will give exception:
  2010-02-26 11:46:12,649: [main] ERROR GrouperStartup.startup(113) - Couldnt startup grouper: java.lang.NullPointerException: at org.hibernate.type.LongType.next(LongType.java:56)
  You should run this script:

```
update GROUPER_ATTRIBUTES set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_ATTRIBUTE_ASSIGN set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_ATTRIBUTE_ASSIGN_VALUE set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_ATTRIBUTE_DEF set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_ATTRIBUTE_DEF_NAME set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_ATTRIBUTE_DEF_NAME_SET set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_ATTRIBUTE_DEF_SCOPE set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_ATTR_ASSIGN_ACTION set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_ATTR_ASSIGN_ACTION_SET set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_AUDIT_ENTRY set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_AUDIT_TYPE set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_CHANGE_LOG_CONSUMER set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_CHANGE_LOG_TYPE set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_COMPOSITES set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_FIELDS set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_GROUPS set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_GROUPS_TYPES set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_GROUP_SET set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_MEMBERS set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_MEMBERSHIPS set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_ROLE_SET set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_STEMS set hibernate_version_number = 0 where hibernate_version_number is null;
update GROUPER_TYPES set hibernate_version_number = 0 where hibernate_version_number is null;
commit;
```

- At this point, your DDL has been upgraded to v1.5.0. However, due to restructing in how we handle effective memberships, there is one additional step that needs to be taken for Grouper to be able to find memberships in your database. Membership objects are now creating by joining the grouper_memberships table with the grouper_group_set table. The latter table is a new table and needs to be populating now. To do so, start up GSH and run the following: new edu.internet2.middleware.grouper.misc.AddMissingGroupSets().addAllMissingGroupSets() For instance..

```
[root@idms-devel-01 grouper]# ./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: 1.5.0, build date: 2009/11/14 13:14:33, env: <no label configured>
grouper.properties read from: /srv/grouper/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: groups@jdbc:oracle:thin:@imsdev-db.oit.duke.edu:1612:IMSDEV
sources.xml read from:        /srv/grouper/conf/sources.xml
sources.xml groupersource id: g:gsa
sources.xml jdbc source id:   jdbc: GrouperJdbcConnectionProvider
Type help() for instructions
gsh 0%
gsh 1%
gsh 2% new edu.internet2.middleware.grouper.misc.AddMissingGroupSets().addAllMissingGroupSets()
```

- Note that when GroupSets are added, memberships are going to appear to the API as if they are new. So if configured, membership hooks will fire, memberships will be added to the changelog and lastMembershipChange (used by LDAPPC) will be updated for groups and stems. To avoid this, simply update the grouper.properties file before adding GroupSets. To disable membership hooks, you can comment out the property hooks.membership.class. To disable the changeLog, set changeLog.enabled to false. And to prevent updates to lastMembershipChange, set groups.updateLastMembershipTime and stems.updateLastMembershipTime to false. After the GroupSets are added, you can set those properties back to the way they were.
- Depending on the number of groups, folders and effective memberships you have, there may be a large number of GroupSets created. One GroupSet is created for each field for each group and stem and one GroupSet is created for each membership where the member is a group. By default, each GroupSet that is created will print a line. For instance, if you have one folder called "etc" and one group called "etc:wheel", you will see output similar to the following.

```
Adding self groupSet for etc for field naming / creators
Adding self groupSet for etc for field naming / stemmers
Adding self groupSet for etc:wheel for field access / viewers
Adding self groupSet for etc:wheel for field access / updaters
Adding self groupSet for etc:wheel for field access / readers
Adding self groupSet for etc:wheel for field access / optins
Adding self groupSet for etc:wheel for field access / optouts
Adding self groupSet for etc:wheel for field access / admins
Adding self groupSet for etc:wheel for field list / members
```

- And for each membership where the member is a group, you will see output similar to the following.

```
Adding groupSet for ownerGroupId = f182a1d4-8bc4-4c96-9249-44b2111ff2a1, memberGroupId = b4450fa0-d746-
4d81-bb5d-7449dc5025b3 for field access / admins
```

- If you would like to prevent output from being printed to your screen, you can call the showResults(boolean) method on AddMissingGroupSets. For instance..

```
gsh 3% new edu.internet2.middleware.grouper.misc.AddMissingGroupSets().showResults(false).
addAllMissingGroupSets()
```

- After you are satisified with the upgrade, there are some backup columns and a backup table that were creating during the registry upgrade that you can remove. If you would like to remove those, do the following.
    - Configure your grouper.properties file to have those backup columns and table dropped.

```
# before the group name etc was moved to the grouper_groups table, the attributes table
# was backed up.  If it should not be backed up, or if the upgrade is done and works, then it can
# be removed, set to true, run: gsh -registry -deep
ddlutils.dropAttributeBackupTableFromGroupUpgrade = true

# Since grouper_memberships no longer has effective memberships, that table doesn't need via_id,
# depth and parent_membership.  If they were converted, this will drop the backup of those cols
with: gsh -registry -deep
ddlutils.dropMembershipBackupColsFromOwnerViaUpgrade = true
```

    - Run gsh -registry -deep For instance..

```
[root@idms-devel-01 grouper]# ./bin/gsh.sh -registry -deep
Using GROUPER_HOME: /srv/grouper
Using GROUPER_CONF: /srv/grouper/conf
Using JAVA: java
using MEMORY: 64m-512m
Grouper starting up: version: 1.5.0, build date: 2009/11/14 14:54:09, env: <no label configured>
grouper.properties read from: /srv/grouper/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: groups@jdbc:oracle:thin:@imsdev-db.oit.duke.edu:1612:IMSDEV
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 or 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
'groups', db url 'jdbc:oracle:thin:@imsdev-db.oit.duke.edu:1612:IMSDEV'? (y|n):
y
Continuing...
Grouper ddl object type 'Grouper' has dbVersion: 0 and java version: 22
Grouper ddl object type 'Subject' has dbVersion: 0 and java version: 1
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/grouperDdl_20091114_15_02_17_992.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/grouperDdl_20091114_15_02_17_992.sql
```

- Check the SQL file to make sure it's okay and then run the SQL file. For instance..

```
[root@idms-devel-01 grouper]# ./bin/gsh.sh -registry -runsqlfile /srv/grouper
/grouperDdl_20091114_15_02_17_992.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 or deny db urls and users in the grouper.properties)
Are you sure you want to run the sql file in db user 'groups', db url 'jdbc:oracle:thin:@imsdev-db.
oit.duke.edu:1612:IMSDEV'? (y|n):
y
Continuing...
Script was executed successfully

Grouper starting up: version: 1.5.0, build date: 2009/11/14 14:54:09, env: <no label configured>
grouper.properties read from: /srv/grouper/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: groups@jdbc:oracle:thin:@imsdev-db.oit.duke.edu:1612:IMSDEV
sources.xml read from:        /srv/grouper/conf/sources.xml
sources.xml groupersource id: g:gsa
sources.xml jdbc source id:   jdbc: GrouperJdbcConnectionProvider
```