Bad Membership Finder

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

Bad Membership Finder Utility

The Bad Membership Finder Utility goes through all of your composite groups and verifies that all composite memberships in the grouper_memberships table have been correctly computed from immediate memberships. As of Grouper v. 2.1, the utility also verifies that composite groups do not have immediate memberships and that non-composite groups do not have composite memberships. Also, as of v. 2.1, there are some groupSet checks to make sure that type field is correct, composite groups do not have groupSets with a depth of 1 and there are not any missing effective groupSets. And as of Grouper v. 2.1.6, there is an additional check to detect deleted groups that are members of other groups so that those memberships can be deleted.

This utility, when run via command line, is read-only; it does not make any membership changes in your database. However, the daemon (see below) will actually fix any problems that are found. Note that in 2.3, the daemon can be configured to run on a schedule. In 2.4, the daemon is automatically configured to run at 1am.

Usage

As of Grouper v 2.1, this utility runs with GSH.

```
$GROUPER_HOME/bin/gsh.sh -findBadMemberships <command line arguments>
```

```
usage: FindBadMemberships -all
 -all           Find bad memberships.

This script will find membership records in the database which are invalid and print them on the screen.  It
will not make any modifications to the Grouper database.
If bad memberships are found, this script will create a GSH script that will correct memberships.

To fix your memberships, complete these steps in the order listed:

1.  Review the GSH script before applying any changes to your database.
2.  Run the GSH script.
3.  Re-run the bad membership finder utility to verify that bad memberships have been fixed.
```

If bad memberships or groupSets are found, the utility will print a message to standard output.

Resolving bad memberships

If bad memberships are found in your Grouper database, a GSH script called findbadmemberships.gsh will be created to help you resolve the issues.

To fix your bad membership, do the following:

1. Review the GSH script before applying any changes to your database.
2. Run the GSH script.
3. Re-run the bad membership finder utility to verify that bad memberships have been fixed

Daemon

For Grouper 2.3+ (with API patch 56), there's a daemon that can find and fix bad/missing memberships. Just enable the job and set a schedule in grouper-loader.properties.

```
otherJob.findBadMemberships.class = edu.internet2.middleware.grouper.misc.FindBadMembershipsDaemon
otherJob.findBadMemberships.quartzCron = 0 0 1 * * ?
```

Query for bad composites

(assumes Grouper 2.0 grouper_memberships_lw_v):

Check for bad complements (tested in oracle):

```
SELECT DISTINCT gcv.owner_group_name, gmlv1.subject_source, gmlv1.subject_id
           FROM grouper_memberships_lw_v gmlv1, grouper_composites_v gcv
          WHERE gcv.composite_type = 'complement'
            AND gmlv1.group_name = gcv.left_factor_group_name
            AND gmlv1.list_name = 'members'
            AND gmlv1.subject_source <> 'g:gsa'
            AND gmlv1.member_id NOT IN (
                   SELECT gmlv2.member_id
                     FROM grouper_memberships_lw_v gmlv2
                    WHERE gmlv2.group_name = gcv.right_factor_group_name
                      AND gmlv2.list_name = 'members'
                      AND gmlv2.subject_source <> 'g:gsa')
            AND gmlv1.member_id NOT IN (
                   SELECT gmlv2.member_id
                     FROM grouper_memberships_lw_v gmlv2
                    WHERE gmlv2.group_name = gcv.owner_group_name
                      AND gmlv2.list_name = 'members'
                      AND gmlv2.subject_source <> 'g:gsa');
```

Check for bad complements part 2

```
SELECT DISTINCT gcv.owner_group_name, gmlv1.subject_source, gmlv1.subject_id
            FROM grouper_memberships_lw_v gmlv1, grouper_composites_v gcv
           WHERE gcv.composite_type = 'complement'
             AND gmlv1.group_name = gcv.right_factor_group_name
             AND gmlv1.list_name = 'members'
             AND gmlv1.subject_source <> 'g:gsa'
             AND gmlv1.member_id IN (
                    SELECT gmlv2.member_id
                      FROM grouper_memberships_lw_v gmlv2
                     WHERE gmlv2.group_name = gcv.owner_group_name
                       AND gmlv2.list_name = 'members'
                       AND gmlv2.subject_source <> 'g:gsa')
```

Check for bad unions

```
SELECT DISTINCT gcv.owner_group_name, gmlv1.subject_source, gmlv1.subject_id
           FROM grouper_memberships_lw_v gmlv1, grouper_composites_v gcv
          WHERE gcv.composite_type = 'union'
            AND gmlv1.group_name = gcv.owner_group_name
            AND gmlv1.list_name = 'members'
            AND gmlv1.subject_source <> 'g:gsa'
            AND gmlv1.member_id NOT IN (
                   SELECT gmlv2.member_id
                     FROM grouper_memberships_lw_v gmlv2
                    WHERE gmlv2.group_name = gcv.left_factor_group_name
                      AND gmlv2.list_name = 'members'
                      AND gmlv2.subject_source <> 'g:gsa')
            AND gmlv1.member_id NOT IN (
                   SELECT gmlv2.member_id
                     FROM grouper_memberships_lw_v gmlv2
                    WHERE gmlv2.group_name = gcv.right_factor_group_name
                      AND gmlv2.list_name = 'members'
                      AND gmlv2.subject_source <> 'g:gsa');
```

Check for bad intersections

```
SELECT DISTINCT gcv.owner_group_name, gmlv1.subject_source, gmlv1.subject_id
           FROM grouper_memberships_lw_v gmlv1, grouper_composites_v gcv
          WHERE gcv.composite_type = 'intersection'
            AND gmlv1.group_name = gcv.owner_group_name
            AND gmlv1.list_name = 'members'
            AND gmlv1.subject_source <> 'g:gsa'
            AND (   gmlv1.member_id NOT IN (
                       SELECT gmlv2.member_id
                         FROM grouper_memberships_lw_v gmlv2
                        WHERE gmlv2.group_name = gcv.left_factor_group_name
                          AND gmlv2.list_name = 'members'
                          AND gmlv2.subject_source <> 'g:gsa')
                 OR gmlv1.member_id NOT IN (
                       SELECT gmlv2.member_id
                         FROM grouper_memberships_lw_v gmlv2
                        WHERE gmlv2.group_name = gcv.right_factor_group_name
                          AND gmlv2.list_name = 'members'
                          AND gmlv2.subject_source <> 'g:gsa')
                );
```

See also

update in Grouper 2.5x