Import-Export

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

This topic is discussed in the "Grouper API - Part 2" training video.

The Grouper registry can be exported to XML and imported back in another database. There are a few variations on this.

- Grouper database migration utility
- Export / import Grouper MySQL database
- Grouper export to a GSH script

XML Import / Export for Grouper

As of v2.1, not all features are exported/imported with this tool. Examples are external subjects, entities, point-in-time auditing, etc. This can be used to make a backup for upgrades, but shouldn't be used to migrate from one version to another during upgrades.

As of Grouper v1.4.0 the invocation of these tools has moved from Ant to gsh (GrouperShell):

Grouper includes XML import / export tools. Exported XML may be used for:

- provisioning to other systems
- reporting
- backups
- switching database backends - including to upgraded schemas (required by new Grouper API versions) in the same database
- moving or syncing a folder of grouper to another environment

Imported XML may be used for:

- loading - adding to or updating existing Stems, Groups and Group Types. Whole or partial Grouper registries can be exported, and subsequently imported at a specified Stem (or the Root Stem if not specified) in the *new* instance.*
- initializing a new, *empty* registry to a known state - useful for demos, testing and system recovery

In general, exported data can be imported into the same Grouper instance it was exported from**, or a different instance. Stems and Groups and Group Types will be created, if not already present, or updated if they already exist (depending on import options provided).

Any tool which can create XML, in the correct format, can be used as a loader.

*To successfully load Subject data, the new Grouper instance must be configured with the same Subject Sources. The export tool does not export Subject registries. Subjects which cannot be resolved will be logged, but otherwise ignored.

**The initial version of the import tool did not maintain system attributes i.e. uuid, date created etc. Now all metadata about the object is kept in sync, though if an object already exists, it will use the existing uuid, not the imported uuid.

Usage

Export:

```
C:\mchyzer\grouper\trunk\grouper\bin>gsh -xmlexport
Using GROUPER_HOME:           c:\mchyzer\grouper\v2_1\grouper\bin\..
Using GROUPER_CONF:           c:\mchyzer\grouper\v2_1\grouper\bin\../conf
Using JAVA:                   java
using MEMORY:                 64m-750m
Usage:
args: -h,            Prints this message
args:
      [-noprompt] filename
e.g.  gsh -xmlexport f:/temp/prod.xml
e.g.  gsh -xmlexport -stems a:b:c,d:e:f f:/temp/prod.xml

  -includeComments,  Put comments about foreign keys in XML
  -stems,            Only include objects in these comma separated stems or object names
  -objectNames,      Only include objects in these comma separated object names or stems
  -excludeAudits,    Put comments about foreign keys in XML
  -noprompt,         Do not prompt user to confirm the export
  filename,          The file to import

C:\mchyzer\grouper\trunk\grouper\bin>gsh -xmlexport whatever.xml
Using GROUPER_HOME: C:\mchyzer\grouper\trunk\grouper\bin\..
Using GROUPER_CONF: C:\mchyzer\grouper\trunk\grouper\bin\../conf
Using JAVA: java
using MEMORY: 64m-512m
This db user 'grouper' and url 'jdbc:mysql://localhost:3306/grouper' are allowed to be changed in the grouper.
properties
Continuing...
Grouper starting up: version: 1.6.0, build date: 2010/02/09 02:24:03, env: <no label configured>
grouper.properties read from: C:\mchyzer\grouper\trunk\grouper\conf\grouper.properties
Grouper current directory is: C:\mchyzer\grouper\trunk\grouper\bin
log4j.properties read from: C:\mchyzer\grouper\trunk\grouper\conf\log4j.properties
Grouper is logging to file: C:\mchyzer\grouper\trunk\grouper\bin\..\logs\grouper_error.log, at min level WARN
for package: edu.internet2.middleware.grouper, based on log4j.properties
grouper.hibernate.properties: C:\mchyzer\grouper\trunk\grouper\conf\grouper.hibernate.properties
grouper.hibernate.properties: grouper@jdbc:mysql://localhost:3306/grouper
sources.xml read from: C:\mchyzer\grouper\trunk\grouper\conf\sources.xml
sources.xml groupersource id: g:gsa
sources.xml jdbc source id: jdbc: GrouperJdbcConnectionProvider
Starting: 163 records in the DB to be exported
DONE: 02:32:54: exported 163 records to: C:\mchyzer\grouper\trunk\grouper\bin\whatever.xml
C:\mchyzer\grouper\trunk\grouper\bin>
```

Import

Note: you should set include/exclude and require groups off in grouper.properties when importing:

```
grouperIncludeExclude.use = false
grouperIncludeExclude.requireGroups.use = false
```

```
C:\mchyzer\grouper\trunk\grouper\bin>gsh -xmlimport
Using GROUPER_HOME:           c:\mchyzer\grouper\v2_1\grouper\bin\..
Using GROUPER_CONF:           c:\mchyzer\grouper\v2_1\grouper\bin\../conf
Using JAVA:                   java
using MEMORY:                 64m-750m
Usage:
args: -h,            Prints this message
args:
      [-recordReport]
      [-noprompt] filename
e.g.  gsh -xmlimport f:/temp/prod.xml

  -recordReport,     Print a file which lists each insert/update
                     In addition to import
  -noprompt,         Do not prompt user to confirm the database that
                     will be updated
  filename,          The file to import

C:\mchyzer\grouper\trunk\grouper\bin>gsh -xmlimport whatever.xml -recordReport
Using GROUPER_HOME: C:\mchyzer\grouper\trunk\grouper\bin\..
Using GROUPER_CONF: C:\mchyzer\grouper\trunk\grouper\bin\../conf
Using JAVA: java
using MEMORY: 64m-512m
This db user 'grouper' and url 'jdbc:mysql://localhost:3306/grouper' are allowed to be changed in the grouper.
properties
Continuing...
Grouper starting up: version: 1.6.0, build date: 2010/02/09 02:24:03, env: <no label configured>
grouper.properties read from: C:\mchyzer\grouper\trunk\grouper\conf\grouper.properties
Grouper current directory is: C:\mchyzer\grouper\trunk\grouper\bin
log4j.properties read from: C:\mchyzer\grouper\trunk\grouper\conf\log4j.properties
Grouper is logging to file: C:\mchyzer\grouper\trunk\grouper\bin\..\logs\grouper_error.log, at min level WARN
for package: edu.internet2.middleware.grouper, based on log4j.properties
grouper.hibernate.properties: C:\mchyzer\grouper\trunk\grouper\conf\grouper.hibernate.properties
grouper.hibernate.properties: grouper@jdbc:mysql://localhost:3306/grouper
sources.xml read from: C:\mchyzer\grouper\trunk\grouper\conf\sources.xml
sources.xml groupersource id: g:gsa
sources.xml jdbc source id: jdbc: GrouperJdbcConnectionProvider
grouper import: reading document: C:\mchyzer\grouper\trunk\grouper\bin\whatever.xml, version: 1.6.0
XML file contains 163 records
02:34:58: Beginning import: database contains 155 records
Ending import: processed 163 records
Ending import: database contains 163 records
Ending import: 8 inserts, 1 updates, and 154 skipped records
DONE: 02:34:59: imported 163 records from: C:\mchyzer\grouper\trunk\grouper\bin\whatever.xml
Wrote record report log to: C:
\mchyzer\grouper\trunk\grouper\bin\grouperImportRecordReport_2010_02_09__02_34_58_685.txt

C:\mchyzer\grouper\trunk\grouper\bin>more C:
\mchyzer\grouper\trunk\grouper\bin\grouperImportRecordReport_2010_02_09__02_34_58_685.txt
Update: Group: 197c460aff064eb6876b63d500c5ee22, etc:userReceiver
Insert: AttributeDefNameSet: 3e6915e7b4f144b38fe7e5143a60c9b4,
Insert: AuditEntry: f7be69a260514b6db7c3982e997cc012
Insert: AuditEntry: e8bc311da27c468281c4d8867305a998
Insert: AuditEntry: de69f0556d4648169b94ffcb7936cf77
Insert: AuditEntry: faa8130871e549e3947f2d3afaeae460
Insert: AuditEntry: f31a5288f8564b2c8e41a5f693a4f914
Insert: AuditEntry: e5a2c9ef662c483691bd92f8e65d1daa
Insert: AuditEntry: f2227db7415e44659f61e1703a02c81c

C:\mchyzer\grouper\trunk\grouper\bin>
```

Summary

Since so many new columns and tables have been added to Grouper especially in 1.5, and since exporting/importing these with the current design would be difficult, we decided to rewrite the Grouper export/import. It will have these differences from the current version

1. Doesn't store the XML document in memory (SAX) for good memory performance
2. Versioned
3. Doesn't manually marshal XML (will use xstream)

4. Will keep logic in beans (more object oriented)
5. Handles all data columns in the database (e.g. uuids). Note, in import will need to lookup the business key to see if there is a different UUID, and maintain the existing UUID if it exists, will not change any UUIDs on import
6. Handles all the new tables (e.g. new attribute framework, though I didn't think we need to import the "set" tables, e.g. groupSet. We can calculate that stuff after import. This is a tradeoff between size of file, speed of import (probably faster to export the "set" tables), and data integrity (probably better to recalc all after import)
7. Sorted output so XML can be diffed, though uuids might make differs thing there are diffs, when really there might not be
8. The export settings will be at the top of the export file (i.e. are exporting the entire registry?)
9. Basic features will be implemented in the first pass, then we can do the advanced features. e.g. we will export all the data as GrouperSystem
10. The XML export is not intended to be used for reporting or provisioning, web services and SQL can be used for that
11. It's not really possible to have a readonly import mode without a huge transaction that will bog down the db
12. Should have good logging (depending on level), and should print to stdout status periodically. i.e. try to say how many records have been processed and how many left to go (will need a preparse for this). For example every 30 seconds.

Notes

- The table: grouper_ddl will not be exported/imported, since that should be tied to the ddl in the schema
- The subject and subjectattribute tables will not be exported/imported, since they aren't really a part of grouper, just the quick start
- Effective membership create and last update date can be calculated from the repository, but will not be in the first pass (these are some of the group_set records). Same for other _set tables like actions, permissions, roles (well, the immediate part of the tables will be exported)
- Privileges (e.g. admin or read on a group) will be exported from the memberships table if they are there. If you are not using the default privilege interface in grouper.properties, then you need to export/import yourself
- It's assumed that composite memberships don't need to be exported since the composite table is exported, and if the composite is added, the API will add the membership data

Issues

- Should audits be handled as they are now, as a separate file? These are in the audit file so they are handled consistently, and foreign keys get translated
- Should change log be exported/imported? Currently assuming no
- Should audit logs be inserted when importing? Assuming will check for uuid of record, if not found, insert
- Assuming that the legacy imports will still work, at least for a little while longer (at least they will work in v1.6). just use -xmlimportold, or -xmlexportold
- Need to update docs

See Also

The XML Import Export, described above, is for importing/exporting the registry as an admin.
There are also import/export procedures available for a group owner to import/export/change the membership of their group.
To learn about that, see the Lite UI Training Video - Part 2