# **Space Details**

Key:	GrouperWG
Name:	Grouper
Description:	Grouper Product & Project Wiki
Creator (Creation Date):	steveo@internet2.edu (Apr 07, 2006)
Last Modifier (Mod. Date):	jbibbee@internet2.edu (Jul 06, 2006)

# **Available Pages**

- Home 釐
  - About Grouper
    - Credits
  - Grouper Product
    - API Building & Configuration
    - Bad Membership Finder Utility
    - Contact Information
    - Custom Group Types, Fields, Attributes, Lists
    - Customising the Grouper UI
      - Grouper UIs
    - Developer's Guide to the Grouper API
    - Getting Started
    - Glossary
      - UI Terminology
    - Grouper Binary Release
    - Grouper Design Guidelines
    - Grouper Loader
    - GrouperShell (gsh)
    - Grouper UI Components
    - Grouper UI Development Environment
    - Grouper Web Services
      - Authentication for Grouper Web Services
      - Grouper Web Services FAQ
    - Import-Export
    - Initializing Administration of Privileges
    - Intro FAQ
    - License
    - Prerequisites
    - Software Download
      - Grouper change log v1.4
      - v1.4.0 Getting Started
      - v1.4.1 Upgrade Instructions from v1.3.\*
      - v1.4 Release Notes
    - Specsheet

- UI Building and Configuration
- UI Customization Guide
- Unresolvable Subject Deletion Utility (USDU)

### Home

This page last changed on Jan 15, 2009 by awest@internet2.edu.

GROUPER: About FAQ Software Documentation Archives Contribute WG Contact

# Welcome to the Grouper Groups Management Toolkit Wiki

Grouper Product	Grouper Project
Open viewing. Editing restricted to the Grouper Developers.	<i>Open viewing. <u>Editing</u> restricted to Grouper</i> <i>Working Group members.</i>
Grouper Product Documentation	The Grouper Working Group
The Grouper Product space, with the latest software and documentation.	The Grouper Project space, with the design and development work of the MACE-led Grouper Working Group.
<ul> <li>New! - <u>Grouper v1.4.0</u> is available.</li> <li>Intended to house the overview and more technical documents regarding the production release of Grouper.</li> <li>for the manager, sysadmin, and applications developer</li> <li>Find the current or previous versions of the Grouper software.</li> <li>http://grouper.internet2.edu: Grouper's official website</li> <li>Grouper Infosheet (PDF)</li> </ul>	<ul> <li>Intended to house items beyond the convenience of the mailing list.</li> <li>Contribute your campus' software, documentation, use cases here.</li> <li>Storage for Member presentations, draft documents, proposals, etc.</li> <li>Return to the Web: Grouper Working Group Home         <ul> <li><u>Minutes</u> - notes from the WG bi-weekly conference calls</li> <li>WG Final Documents - link coming soon!</li> </ul> </li> </ul>

#### **Request Editing Permissions**

For editing access within the Grouper space, you will need to first obtain a registered Confluence username/password:

- 1. Please sign up with your email as your username, and
- 2. Send an access request to Steve Olshansky <steveo AT internet2 DOT edu>.

### Background

As a result of initial investigations by the MACE-Dir-Groups, Grouper was developed as an open source toolkit to address the needs of managing groups. Grouper is designed to function as the core element of a common infrastructure for managing group information across integrated applications and repositories. Grouper combines multiple sources of group information, both automated and manual, in managing memberships and other group information in a Groups Registry, a central information asset complementary to a site's Person Registry.

A few of the benefits of a groups management service, such as Grouper, include:

- the same groups are made available to many applications
- · distributed authorities are able to directly manage access information
- sophisticated group management capabilities, such as subgroups and composite groups, to support many access management needs
- common user, web services, command line, and java interfaces for managing groups

Grouper supports several modes of distributed group management:

- per-group assignment of membership update privilege
- per-group assignment of administration of all forms of access
- per-naming stem assignment of entitlement to create groups within each namespace
- per-group opt-in or opt-out entitlement

NOTE WELL: All Internet2 Activities are governed by the Internet2 Intellectual Property Framework.

#### Working Group Chair

#### Working Group Flywheel

Tom Barton, University of Chicago

Steve Olshansky, Internet2

#### Mailing Lists

To subscribe to Grouper mailing lists, including Grouper-Announce, see the <u>Contact</u> page.

#### **Related Internet2 Middleware projects**

- <u>Signet</u>
- <u>Shibboleth</u>

Development of this software was supported with funding from <u>Internet2</u>, the <u>University of Chicago</u>, the <u>University of Bristol</u>, the <u>NSF Middleware Initiative</u> (NSF 02-028, Grant No. OCI-0330626), and <u>JISC.</u>Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the <u>National Science Foundation</u> (NSF).

Questions or comments? <sup>1</sup> Contact us.

<u>GROUPER: About</u> FAQ Software Documentation Archives Contribute WG Contact Comments

Getting another ID here is a hassle. I already have a Shib ID from ProtectNetwork. Why cant you guys support ProtectNetwork Shib ID or InCommon or SDSS on this site? Thanks.

Posted by at Sep 18, 2006.

HI, I would not have thought it should be that hard - I had no problems

James http://www.software-dungeon.co.uk

Posted by at Mar 27, 2007.

## **About Grouper**

This page last changed on Dec 06, 2007 by jbibbee@internet2.edu.

GROUPER: About FAQ Software Documentation Archives Contribute WG Contact

# Why is Grouper Open Source?

The Grouper product is an open-source project, and is a result of the efforts by the Groups subgroup of <u>MACE-Dir</u>. Grouper aims to satisfy the need for need a collaborative groups management tool, integrating and enabling authenticated access to groups data from applications and repositories across an institution.

Grouper has been under development since 2001, under the guidance of the MACE-Dir-Group and efforts of the development team and working group members.

Grouper is now at a stage fit for production-level use, though the project will continue to evolve and benefit from contributions of the users. The Working Group welcomes all ideas towards the design aspects of functionality and deployment. Any changes and improvements will be a direct result of collaborations between the user-base and the developers. Your continued support of Grouper will further enhance the efforts to date.

- Licensed under the Apache 2.0 license.
- Grouper Credits

Internet2 Contributor Software License Agreements: <u>http://members.internet2.edu/</u> intellectualproperty.html#appendix\_c

### Thanks!

The Grouper team would like to thank all who have contributed to the development of Grouper; in particular, the MACE-Dir-Groups Working Group members, Tom Barton (MACE-Dir-Groups Working Group chair) and Blair Christensen of the University of Chicago, and Gary Brown of the University of Bristol, who have contributed their time, expertise, and enthusiasm for this project: also Jessica Bibbee, Nate Klingenstein, Liene Karels, Renee Frost, Ann West, and Steve Olshansky from Internet2.

Development of this software was supported with funding from <u>Internet2</u>, <u>University of Chicago</u>, <u>University of Bristol</u>, the <u>NSF Middleware Initiative</u> (NSF 02-028, Grant No. OCI-0330626), and <u>JISC</u>. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the <u>National Science Foundation</u> (NSF).

Questions or comments? 1 Contact us.

**<u>GROUPER:</u>** <u>About</u> <u>FAQ</u> <u>Software</u> <u>Documentation</u> <u>Archives</u> <u>Contribute</u> <u>WG</u> <u>Contact</u>

# Credits

This page last changed on Aug 12, 2007 by steveo@internet2.edu.

**<u>GROUPER</u>:** About FAQ Software Documentation Archives Contribute WG Contact

# **Grouper Credits**

The credits page is now linked from the <u>About</u> page or direct at <u>http://</u> middleware.internet2.edu/dir/groups/grouper/credits.html

**Questions or comments? Contact us.** 

GROUPER: About FAQ Software Documentation Archives Contribute WG Contact

## **Grouper Product**

This page last changed on Jan 27, 2009 by tbarton@uchicago.edu.

Contact us if you have additional comments or suggestions.

**<u>GROUPER</u>: About FAQ Software Documentation Archives Contribute WG Contact</u>** 

# Welcome to the Grouper v1.4.0 Product Documentation Space

**NEW!!** Grouper v1.4.0 is now available!

# **Software Overview**

Download - Download the latest product version from the main Grouper Software web site.
 Specsheet - Offers operational specifications for the development and deployment of Grouper.
 License - Terms under which Grouper is licensed, the Apache 2.0 license.
 Archives - Details feature information and downloads of previous Grouper versions.
 Glossary - Terms and definitions relevant to Grouper.

Grouper Trademark Guidelines and Styleguide - Usage policy for the Grouper logo.

# **Grouper Integration**

Graphic illustration of the the range of options for integrating Grouper with existing Identity Management components and with applications.



# **Systems Administration**

#### **Installation & Configuration**

<u>Prerequisites</u> - Establish the Grouper environment. <u>API Installation</u> - Configure the API and integrate with existing identity stores. <u>UI Installation</u> - Configure, build, and deploy the Grouper User Interface. <u>WS Installation</u> - Build, secure, and deploy Grouper Web Services. <u>Initializing Administration of Privileges</u> - The first stem and group you should create.

#### **Tools & Topics for On-Going Administration**

<u>GrouperShell</u> - Documentation for the *gsh* command line utility. <u>Import/Export Tool</u> - Documentation for the XML Import/Export tool. <u>Unresolvable Subject Deletion Utility</u> - Documentation for the command line *usdu* tool. <u>Bad Membership Finder Utility</u> - Documentation for the *findbadmemberships* command line script. <u>Grouper Client</u> - Experimental client for Grouper LDAP and Web Services.

<u>Custom Group Types & Fields</u> - What they are and how to create and delete them.

# **Applications Development**

<u>Grouper UI Customisation Guide</u>\*- click here for more information regarding the following documents:

- Grouper UI Components
- Architecture
- Struts Actions and Tiles
- Customising the Grouper UI
- Grouper UI Development Environment

Developer's Guide to Grouper Web Services

<u>API & UI v1.4.0 Javadoc</u> <u>WS v1.4.0 Javadoc</u> Grouper Client v1.4.0 Javadoc

<u>CVS</u> - manages the versioning of the Grouper source code. Access the Grouper source code via the web:

- <u>Connection type</u>: pserver
- User: anoncvs
- Passwd: <your email address>
- Host: anoncvs.internet2.edu
- Repository Path: /home/cvs/i2mi
- Use default port: yes
- · Access the complete Grouper source code via the command line:

```
cvs -z3 -d :pserver:anoncvs@anoncvs.internet2.edu:/home/cvs/i2mi login
cvs -z3 -d :pserver:anoncvs@anoncvs.internet2.edu:/home/cvs/i2mi co -r GROUPER_1_4_0 grouper
cvs -z3 -d :pserver:anoncvs@anoncvs.internet2.edu:/home/cvs/i2mi co -r GROUPER_UI_1_4_0
grouper-ui
cvs -z3 -d :pserver:anoncvs@anoncvs.internet2.edu:/home/cvs/i2mi co -r GROUPER_WS_1_4_0
grouper-ws
```

Bug Reports - Bugs submitted and fixes found here. Note: You will need to create a login with Jira.

# **Internet2 Middleware Initiative Commons**

This area will house documents and other items that are shared between the projects of the Internet2 Middleware Initiative (I2MI). See also <u>I2MI-Common in CVS.</u>

**Subject API**- integrates with existing sources of identities whose memberships or privileges are to be managed.

**Ldappc** - automates the reflection of group, membership, and permission information contained in the Groups and Privileges Registries into a site's LDAP directory service.

**Roadmap for I2MI Common Products** - notable future enhancements to the I2MI-Common products.

**<u>Combined Roadmap</u>** for I2MI Common, Grouper, and Signet products.

# **Archived Documentation**

Archived documentation can be viewed on the <u>Archives</u> page, bundled in a (.PDF) file under each release version (major- and sub-releases only.)

# **Community Area**

**Documents & Presentations** - View others' and post your Grouper-related drafts and final works.

**<u>Contributions</u>** - Share your code, software, documentations, use cases, and other contributions with the Grouper community.

**Ontact us** if you have additional comments or suggestions.

**<u>GROUPER</u>: <u>About</u>** <u>FAQ</u> <u>Software</u> <u>Documentation</u> <u>Archives</u> <u>Contribute</u> <u>WG</u> <u>Contact</u> Comments

Those looking to get the latest cvs head for grouper the suitbale cvs invocation seems to be:

cvs -z3 -d :pserver:anoncvs@anoncvs.internet2.edu:/home/cvs/i2mi login

cvs -z3 -d :pserver:anoncvs@anoncvs.internet2.edu:/home/cvs/i2mi co -r HEAD grouper

Posted by calebracey at Oct 10, 2007.

# **API Building & Configuration**

This page last changed on Feb 02, 2009 by tbarton@uchicago.edu.

GROUPER: About FAQ Software Documentation Archives Contribute WG Contact

# Building the Grouper API as of v1.4.0

As of Grouper v1.4.0, the API is now provided both as a binary and source distribution.

To build the source distribution :

cd grouper-api-1.4.0 ant dist

Testing the API is performed using GrouperShell :

Testing will destroy any pre-existing data in the Groups Registry database

bin/gsh.sh -test -all

# Configuring the Grouper API as of v1.4.0

In this section we describe all of the Grouper API configuration files and important settings.

The Grouper API is distributed with example configuration files with ".example" inserted in the middle of their names. These should be renamed or copied to remove the ".example" substring, or doing a build with ant will do this, or it is already copied in the binary distribution. e.g. for grouper.properties, the example file is grouper.example.properties.

Section	Configuration File	Purpose
Database-Related Settings and	grouper.hibernate.properties	integrating the Grouper API with
Procedures		the database that will house your
		Groups Registry
Configuration of Source Adapters	sources.xml	integrating the Grouper API with
-		chosen identity sources
Grouper properties	grouper.properties	defaults for Grouper privileges,
		enabling identified external users
		to act with elevated root-like
		privilege, changing the display
		name for internal subjects
Logging	log4j.properties,	logging
	grouper.properties	
Loader	grouper-loader.properties	auto-load memberships from
		external sql sources. or run other
		jobs

# **Database-Related Settings and Procedures**

#### **Database Driver Location**

Place the jar file containing the JDBC driver for your database in the lib/custom/ directory. The Grouper v1.4 package includes the JDBC driver for HSQLDB v1.7.2.11. Sample JDBC drivers are located in lib/ jdbcSample (e.g. for Oracle, MySQL, and PostgreSQL).

#### **General Property Settings**

Grouper uses Hibernate to persist objects in the Groups Registry. Database-specific settings are configured in conf/grouper.hibernate.properties, which has pre-populated examples for HSQLDB, MySQL, Oracle, and Postgresql.

#### Required properties are:

Property Name	Purpose
hibernate.connection.driver_class	JDBC driver classname
hibernate.connection.url	JDBC URL for the database
hibernate.connection.username	database user
hibernate.connection.password	database user's password Note, you can also put a
	filename of the encrypted password
hibernate.dialect	classname of a Hibernate dialect, for setting platform specific features. Choices are listed <u>here</u>

You may need to refer to your database support person to determine these required properties.

Detailed Hibernate configuration documentation is available here.

#### **MySQL Transaction Support**

If you want transactions to work (i.e. when doing a unit of work in grouper, it either all completes or none), which is definitely recommended, though not required, your mysql table format needs to be transactional, e.g. innodb, which is not the default (myisam is the default). One way to enable innodb in mysql is with this line in the my.cnf: default-storage-engine=innodb

#### **Database Whitelists and Blacklists**

Some database operations (such as dropping tables or recreating data during tests) require confirmation of a prompt asking whether or not to continue. It is possible to automatically allow or deny these database operations in conf/grouper.properties :

# whitelist (allow) and blacklist (deny) for db data or object deletes. # if a listing is in the whitelist (allow), it will be allowed to delete db # if a listing is in the blacklist (deny), it will be denied from deleting db # multiple inputs can be entered with .0, .1, .2, etc. These numbers must be sequential, starting with 0 db.change.allow.user.0=grouper3 db.change.allow.url.0=jdbc:mysql://localhost:3306/grouper3 db.change.allow.user.1=grouper1 db.change.allow.url.1=jdbc:mysql://localhost:3306/grouper1

db.change.deny.user.0=grouper2
db.change.deny.url.0=jdbc:mysql://localhost:3306/grouper2

#### **Database Initialization Procedure**

Database initialization is performed using the GrouperShell.

Initializing the database will destory any pre-existing data

To initialize the Groups Registry and install tables, populate default group types and fields, and create the root naming stem :

bin/gsh.sh -registry -check -runscript

To re-initialize the Groups Registry (e.g. after running junit tests) :

bin/gsh.sh -registry -reset

To see all options :

bin/gsh.sh -registry

# **Configuration of Source Adapters**

Grouper uses <u>Subject API</u> compliant "source adapters" to integrate with external identity stores. "Subjects" are the objects housed there that are presented to Grouper for management vis-à-vis group membership and Grouper privileges. These may represent people, other groups, computers, applications, services, most anything for which you manage identity. With the exception of Grouper groups, Grouper treats all subjects opaquely. See the <u>Subject API</u> documentation for further background and details concerning subjects, source adapters, and other aspects of the Subject API.

Each source adapter connects with a single back-end store using JDBC or JNDI. Grouper makes no specific assumptions about the schema of any subject types. Instead, sections of the configuration file, grouper/conf/sources.xml, declare the details of how to connect with each back-end store, the identifier(s) to be used for the subjects it contains, how to select and search for subjects, and which subject attributes should be made available to Grouper.

Grouper v1.4.0 relies on v0.4.2 of the Subject API. Please refer to the section on <u>Subject API v0.3.1</u> for detailed configuration information.

Three source adapter classes are included in the Grouper API v1.3.0 package. JDBCSourceAdapter and JNDISourceAdapter classes are included in subject-0.3.1.jar, and GrouperSourceAdapter is built along with the Grouper API. Every Grouper API deployment MUST include a\*source\* element in grouper/conf/ sources.xml for the GrouperSourceAdapter so that Grouper can refer to its own groups in the same manner as other subjects.

See the sources.example.xml for example usages of the sources.xml

#### **Choosing Identifiers for Subjects**

Identifiers and their management can get complicated. They can be revoked or not, re-assigned or not, lucent or opaque, etc. Depending on such characteristics, a given identifier might be a good or bad choice to use in the context of managing the identified subject's group memberships.

For example, a username is often lucent - easily remembered by the person to whom it is associated. But it may also be revokable, meaning that it no longer refers to that person (perhaps they have a new one), or even re-assignable, meaning that it might refer to some other person at a later time. If a username is used to record membership, username changes must trigger corresponding membership changes. A username is better suited to authentication than it is to indicating membership.

On the other hand, an opaque registryID (machine, not human, readable) that never changes is great for membership, but lousy for authentication - it might not even be known by the person to whom it is associated. How would I identify myself to Grouper if I wished to opt-in to a list or manage a group?

Grouper accommodates subject identifier issues in two ways. First, it maintains UUIDs for every subject and group within the Groups Registry. These are never exposed by the API, but are associated with externally supplied subject identifiers within the Groups Registry. This approach allows the identifier associated with a given subject to be changed without any need to change actual memberships.

Second, by relying on the Subject API, Grouper is able to lookup subjects that are presented with an identifier in one namespace and obtain identifiers in other namespaces for that subject. That means that it can translate a username into a registryID, for example. So, when a user authenticates to an application using the Grouper API, that application can use the Subject API to fetch an identifier for the person chosen by the site for use in memberships. Similarly, when a membership in the Groups Registry is to be expressed elsewhere, the identifier used for group members can be translated by a provisioning connector by use of the Subject API into one that is suitable in the provisioned context.

# **Grouper Properties**

All configuration of Grouper proeprties detailed in this section occur in the grouper/conf/ grouper.properties file. Look in the grouper.example.properties file for the more obscure settings. Common settings are listed below

This setting describes the env that grouper is running, e.g. used in the daily report from the loader which

grouper.env.name = production

If grouper should auto init the registry if not initted (i.e. insert the root stem, built in fields, etc)

registry.autoinit = true

If grouper should try and detect and log configuration errors on startup, in general this should be true, unless the output is too annoying or if it is causing a problem

configuration.detect.errors = true

If groups like the wheel group should be auto-created for convenience (note: check config needs to be on)

configuration.autocreate.system.groups = false

Auto-create groups (increment the integer index), and auto-populate with users (comma separated subject ids) to bootstrap the registry on startup (note: check config needs to be on). The next group would end in 1, then 2, etc

configuration.autocreate.group.name.0 = etc:uiUsers configuration.autocreate.group.description.0 = users allowed to log in to the UI configuration.autocreate.group.subjects.0 = johnsmith

By default, anyone with admin rights on a group can edit the types or attributes. Specify types (and related attributes) which are wheel only, or restricted to a certain group

security.types.typeName.wheelOnly = true
security.types.grouperLoader.wheelOnly = true

#security.types.typeName.allowOnlyGroup = etc:someAdminGroup

If you don't want to be prompted for DDL changes in certain databases (e.g. dev), list them here: Whitelist (allow) and blacklist (deny) for db data or object deletes, without prompting the user to confirm If a listing is in the whitelist (allow), it will be allowed to delete db If a listing is in the blacklist (deny), it will be denied from deleting db Multiple inputs can be entered with .0, .1, .2, etc. These numbers must be sequential, starting with 0

db.change.allow.user.0=grouper3
db.change.allow.url.0=jdbc:mysql://localhost:3306/grouper3
db.change.allow.user.1=grouper1
db.change.allow.url.1=jdbc:mysql://localhost:3306/grouper1

db.change.deny.user.0=grouper2
db.change.deny.url.0=jdbc:mysql://localhost:3306/grouper2

There is a substantial section for <u>include/exclude and requireGroups</u>. These are group types which help you create composite groups to manage include/exclude lists for groups (especially useful for grouper loader privisioned groups), or groups which require memberships in other groups (e.g. activeStaff). See the grouper.example.properties file if you want to customize things, but to enable, set these:

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

Here are some requireGroups (increment the 0 to add more):

#grouperIncludeExclude.requireGroup.name.0 = requireActiveStudent #grouperIncludeExclude.requireGroup.attributeOrType.0 = attribute #grouperIncludeExclude.requireGroup.group.0 = school:community:activeStudent #grouperIncludeExclude.requireGroup.description.0 = If value is true, members of the overall group must be an active student (in the school:community:activeStudent group). Otherwise leave this value not filled in.

<u>Hooks</u> are ways to plugin in your own java code to affect how Grouper does its logic. You can register multiple classes for one hook base class by comma separating the hooks implementations. You can also register hooks at runtime with: GrouperHookType.addHookManual("hooks.group.class", YourSchoolGroupHooks2.class);

See the grouper.example.properties for the full list, here are two examples:

#implement a group attribute hook by extending
edu.internet2.middleware.grouper.hooks.AttributeHooks
#hooks.attribute.class=edu.yourSchool.it.YourSchoolGroupHooks,edu.yourSchool.it.YourSchoolGroupHooks2

#implement a group hook by extending edu.internet2.middleware.grouper.hooks.GroupHooks

#hooks.group.class=edu.yourSchool.it.YourSchoolGroupHooks,edu.yourSchool.it.YourSchoolGroupHooks2

# You can validate group attributes via regex (see grouper.example.properties for more info) (increment the 0 to add more)

#group.attribute.validator.attributeName.0=extension #group.attribute.validator.regex.0=^[a-zA-Z0-9]+\$ #group.attribute.validator.vetoMessage.0=Group ID '\$attributeValue\$' is invalid since it must contain only alpha-numerics

#### Database structure data definition language (DDL) settings (see grouper.example.properties for full list)

# if you want to not create the subject tables (grouper examples for unit testing), # then set this to true ddlutils.exclude.subject.tables = false # set the path where ddl scripts are generated (they will be uniquely named in this directory). # if blank, the directory used will be the current directory ddlutils.directory.for.scripts = ddlScripts # during schema export, should it install grouper data also or not. e.g. insert the root stem, default true ddlutils.schemaexport.installGrouperData = true # when grouper starts, should it shut down if not right version? ddlutils.failIfNotRightVersion = true

# after you have converted id's, and are happy with the conversion of removing the uuid col, # this will remove the backup uuid cols when running the gsh command: registryInitializeSchema() ddlutils.dropBackupUuidCols = false

# after you have converted field id foreign keys, and are happy with the conversion of removing the attribute name,

# membership list name, and type cols, # this will remove the backup field name/type cols when running the gsh command: registryInitializeSchema() ddlutils.dropBackupFieldNameTypeCols = false

# this is the schema ddlutils uses to query metadata with jdbc. usually this can be omitted, # and it defaults to your database loginid, however, in postgres, it can be different, so enter here #ddlutils.schema = public

#if you are running a DB that supports them, but you dont want them, disable views or object comments here (defaults to false) ddlutils.disableComments = false ddlutils.disableViews = false

#### Mail settings (optional, e.g. for daily report form loader)

#smtp server is a domain name or dns name, must be simple clear text stmp with no authentication
#mail.smtp.server = whatever.school.edu

#leave blank if unauthenticated
#mail.smtp.user =
#leave blank if unauthenticated
#mail.smtp.pass =
#this is the default email address where mail from grouper will come from
#mail.from.address = noreply@school.edu
#this is the subject prefix of emails, which will help differentiate prod vs test vs dev etc
#mail.subject.prefix = TEST:

#### **Default privileges**

Grouper requires that all subjects must be explicitly granted access or naming privileges (cf. <u>Glossary</u>), with one caveat. There is a special "subject" internal to Grouper called the ALL subject, which is a standin for any subject. The ALL subject can be granted a privilege in lieu of assigning that privilege explicitly to each and every subject.

When a new group or naming stem is created, any of its associated privileges can be granted by default to the ALL subject. This is configured by a series of properties in grouper.properties, one per privilege. If a property has the value "true" then ALL is granted that privilege by default when a group or naming stem is created. Otherwise it is not, and hence no subject has that privilege by default.

Property Name	Value in Grouper v1.3.1 Distribution
groups.create.grant.all.admin	false
groups.create.grant.all.optin	false
groups.create.grant.all.optout	false
groups.create.grant.all.update	false
groups.create.grant.all.read	true
groups.create.grant.all.view	true
stems.create.grant.all.create	false
stems.create.grant.all.stem	false

#### Super-user Privileges

Grouper has another special "subject" called GrouperSysAdmin that acts as a super-user. GrouperSysAdmin is permitted to do everything - the privilege system is ignored for that special subject. Grouper can be configured to consider all members of a distinguished group to be able to act as superusers, much as the "wheel" group does in BSD Unix. Two properties control this behavior:

Property Name	Description
groups.wheel.use	"true" or "false" to enable or disable this capability.
groups.wheel.group	The group name of the group whose members are to be considered security-equivalent to GrouperSysAdmin. By default this is: etc:sysadmingroup

The Grouper UI enables users that belong to the wheel group to choose when to act with the privileges of GrouperSystem and when to act as their normal selves.

#### Changing the display name of GrouperAll and GrouperSystem

Before version 1.3.0 the Grouper UI referred to *EveryEntity* as GrouperAll and *GrouperSysAdmin as* GrouperSystem. As of version 1.3.1 the name attribute of GrouperAll and GrouperSystem can be set through the properties below.

Property Name	Description
subject.internal.grouperall.name	The name to use for GrouperAll instead of EveryEntity
subject.internal.groupersystem.name	The name to use for GrouperSystem instead of GrouperSysAdmin

If you choose not to use the defaults you will have to update the UI nav.properties file to ensure consistency e.g. subject.privileges.from-grouperall=inherits from EveryEntity

#### Changing default privilege caching

Grouper includes three 'PrivilegeCache' implementations:

- NoCachePrivilegeCache No caching performed
- SimplePrivilegeCache Caches results but flushes all cached entries upon any update
- SimpleWheelPrivilegeCache Same as 'SimplePrivilegeCache' but with better support for using a wheel group.

The privileges.access.cache.interface and privileges.naming.cache.interface properties can be set to determine the privilege caching regimen. The default is edu.internet2.middleware.grouper.NoCachePrivilegeCache.

#### Using a privilege management system external to Grouper

Grouper's internal security implementation relies on two java interfaces, one for Naming Privileges and another for Access Privileges. Grouper ships with classes that implement these interfaces, but 3rd parties are free to supply their own and so manage Grouper privileges using a privilege management system external to Grouper. Two properties declare the java classes that Grouper will use to implement these interfaces:

Property Name	Description
privileges.access.interface	classname of the java class that implements the
	Access Interface
privileges.naming.interface	classname of the java class that implements the Naming Interface

It is not clear that this has been taken advantage of... the internal privilege management is the one usually used

# Logging

Logging is configured in the grouper/conf/log4j.properties configuration file. By default Grouper will write event log information to grouper/grouper\_event.log, error logging to grouper/grouper-error.log, and debug logging, if enabled, to grouper/grouper-debug.log. The log4j configuration can be adjusted to control the verbosity, type and output of Grouper's logging.

In addition, there are several configuration parameters in grouper/conf/grouper.properties that may be adjusted to control the logging of effective membership modifications in the event log.

```
# Control whether the addition and deletion of effective groups memberships
# are logged in the event log. If using the _GrouperAccessAdapter_ this
# will include granted and revoked access privileges.
memberships.log.group.effective.add = true
memberships.log.group.effective.del = true
# If using _GrouperNamingAdapter_, control whether the granting and
# revoking of effective naming privileges are logged in the event log.
memberships.log.stem.effective.add = true
memberships.log.stem.effective.del = true
```

# Loader

The <u>Grouper - Loader</u> is a daemon command line process which run jobs that load/remove memberships of groups based on results from a sql query. There is a grouper-loader.properties file to configure. The common settings are described below, see the grouper-loader.example.properties for descriptions of all settings.

If you want grouper to make sure the loader type and attributes exist if not there, set this. Otherwise you need to add the type and attributes yourself with GSH.

# auto-add grouper loader types and attributes when grouper starts up if they are not there loader.autoadd.typesAttributes = false

If most of your loader queries come from one subject source, you can set the default subject source here, so your loader queries only need to return SUBJECT\_ID and not the source id also:

default.subject.source.id =

If all of your queries are run against your grouper db credentials (e.g. if you have other schemas on the same DB to query), you dont have to configure DB connections. If you have other databases to query

(e.g. an external data warehouse, etc), you can configure the db credentials here. The name here is "warehouse", use different names for different connections

db.warehouse.user = mylogin #note the password can be stored encrypted in an external file db.warehouse.pass = secret db.warehouse.url = jdbc:mysql://localhost:3306/grouper db.warehouse.driver = com.mysql.jdbc.Driver

If you want to use the Grouper daily report, configure in the grouper-loader.properties (and the mail settings described above in grouper.properties). This is a daily email that gets sent to you about your grouper health, including status about all the loader jobs in the last day.

```
#quartz cron-like schedule for daily grouper report, the default is 7am every day: 0 0 7 * * ?
#leave blank to disable this
daily.report.quartz.cron =
```

#comma separated email addresses to email the daily report, e.g. a@b.c, b@c.d
daily.report.emailTo =

Questions or comments? <sup>1</sup> Contact us.

GROUPER: About FAQ Software Documentation Archives Contribute WG Contact

# **Bad Membership Finder Utility**

This page last changed on Jan 04, 2009 by tbarton@uchicago.edu.

**<u>GROUPER</u>:** <u>About</u> <u>FAQ</u> <u>Software</u> <u>Documentation</u> <u>Archives</u> <u>Contribute</u> <u>WG</u> <u>Contact</u>

# **Bad Membership Finder Utility**

This document is released alongside Grouper v1.4.0.

The Bad Membership Finder Utility goes through all of your list memberships, access privileges and naming privileges stored in the grouper\_memberships table and finds any bad effective and composite memberships that have been incorrectly generated from immediate memberships.

This utility is divided into three phases.

Phase 1: The first phase will locate memberships that are associated with a group or stem that no longer exists.

Phase 2: The second phase will locate bad list memberships and access privileges for groups. Phase 3: The third phase will locate bad naming privileges for stems.

Note that because this utility has to go through all of your groups and stems, it may take several hours to run if you have a large (1 million or more) number of memberships.

This utility is read-only; it does not make any membership changes in your database.

#### Usage

As of Grouper 1.4.0, this utility runs with GSH.

\$GROUPER\_HOME/bin/gsh.sh -findBadMemberships <command line arguments>

usage: FindBadMemberships -all | -group <arg> | -stem <arg> -all Find bad list memberships, access privileges, and naming privileges owned by all groups and stems. -group <arg> Find bad list memberships and access privileges for a specific group. -stem <arg> Find bad naming privileges for a specific stem.

This script will find bad effective and composite memberships in your Grouper database. It will not make any modifications to the Grouper database. If bad memberships are found, this script will create a GSH script that will delete and re-add 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 for all groups and stems since additional membership errors for other groups and stems may be revealed after the current groups and stems are fixed.

For every bad membership, the utility will print one line.

#### If the membership does not belong to any group or stem, you will see a message like the following:

FOUND BAD MEMBERSHIP: Membership with uuid=5de0b45e-flbe-442d-8240-9aacc4b1054c has invalid owner with uuid=5423131e-667c-4463-8e85-e5d16864f3ab.

Otherwise, you will see a message like the following for bad memberships with a group:

FOUND BAD MEMBERSHIP: Bad membership in group with uuid=c59c0b99-a735-4798-841a-a497d31afd0b and name=i2:test.

And you will see a message like the following for bad memberships with a stem:

FOUND BAD MEMBERSHIP: Bad membership in stem with uuid=c59c0b99-a735-4798-841a-a497d31afd0b and name=i2.

# **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. For each group with a bad membership, the following will be added to the GSH script.

- 1. delMember() commands for each immediate member of the group.
- 2. revokePriv() commands for each access privilege.
- 3. A delComposite() command if this group is a composite group.
- 4. SQL statements to delete any remaining effective and composite memberships.
- 5. addMember() commands to re-add immediate members.
- 6. grantPriv() commands to re-add access privileges.
- 7. addComposite() to re-create the composite if this is a composite group.

For each stem with a bad membership, the following will be added to the GSH script.

- 1. revokePriv() commands for each naming privilege.
- 2. SQL statements to delete any remaining effective and composite memberships.
- 3. grantPriv() commands to re-add naming privileges.

For each membership with an invalid owner, the following will be added to the GSH script.

1. An SQL statement to simply delete the membership.

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 for all groups and stems since additional membership errors for other groups and stems may be revealed after the current groups and stems are fixed.

Note that you should try to avoid membership updates to the bad groups and stems between the time the GSH script is created and the time that you execute the script. If updates are made to the groups and stems between this time, you should recheck the groups and stems using findBadMembershipsByGroup(group) and findBadMembershipsByStem(stem).

Also note that depending on the type of bad membership found, your execution of the GSH script may produce a constraint violation due to the foreign key fk\_memberships\_parent. If this happens, executing the GSH script a second time should succeed in most cases. But if it doesn't succeed, drop the foreign key, execute the GSH script, and then add the foreign key back.

🕖 Questions or comments? 🟮 Contact us.

**<u>GROUPER</u>: About FAQ Software Documentation Archives Contribute WG Contact</u>** 

# **Contact Information**

This page last changed on Jul 18, 2007 by jbibbee@internet2.edu.

**<u>GROUPER:</u>** <u>About</u> <u>FAQ</u> <u>Software</u> <u>Documentation</u> <u>Archives</u> <u>Contribute</u> <u>WG</u> Contact

**E-Mail Us -** If you have questions or comments about the wiki, documentation, and/or the Grouper project, please email us: grouper-info AT internet2 DOT edu .

# **Support & Resources**

#### **Grouper Working Group**

If you are interested in or have questions regarding the development of Grouper, please visit the <u>Grouper</u> <u>Working Group</u> web or <u>GrouperWG</u> wiki.

#### **Grouper Mailing Lists**

Below is a description of the mailing lists you can join to assist your deployment of Grouper.

To subscribe to any of these lists, send email to sympa AT internet2 DOT edu with the following in the \*subject line\*:

subscribe <list name> <your name>

For example:

#### subscribe grouper-dev Jane Doe

Grouper-Developers Mailing List: Sites wishing to further participate in the development of Grouper and contribute work towards these efforts are encouraged to join the <grouper-dev AT internet2 DOT edu> mailing list. A bi-weekly Grouper (dev) Working Group conference call is held in discussion of development efforts.

#### subscribe grouper-users Jane Doe

Questions and comments regarding the implementation of Grouper should be directed to this list. Messages will be archived, and can be accessed for reference. It is anticipated that subsequent discussion will be supported by both the Grouper developers *and* implementing sites. Your contributions here - comments, questions, and concerns - will be of great value to the general Grouper community. This should include, but not be limited to, questions about the documentation, undocumented problems, installation or operational issues, and other product issues that arise.

#### subscribe mw-announce Jane Doe

Periodic news and announcements about Grouper and other items of broad interest, such as pointers to new standards or discussion lists, major updates to widely-used middleware tools, and information about white papers and best-practices documents. Posting to mw-announce is done by Internet2 staff; to have an item considered for posting, send it to mw-announce-submit AT internet2 DOT edu

To **unsubscribe** from any of these lists, send email to sympa AT internet2 DOT edu with the subject: unsubscribe grouper-dev

unsubscribe grouper-users unsubscribe grouper-announce

Archive: <u>grouper-dev list</u> Archive: <u>grouper-users list</u> Archive: <u>mw-announce list</u> Archive: <u>i2mi-ui list</u> - Internet2 Middleware Initiative SW User Interface Development list

🥑 Questions or comments? 🔨 <u>Contact us</u>.

GROUPER: About FAQ Software Documentation Archives Contribute WG Contact

# **Custom Group Types, Fields, Attributes, Lists**

This page last changed on Apr 25, 2008 by mchyzer.

**<u>GROUPER</u>:** <u>About</u> <u>FAQ</u> <u>Software</u> <u>Documentation</u> <u>Archives</u> <u>Contribute</u> <u>WG</u> <u>Contact</u>

This document will explain how to work with custom group types, and how you can best apply it to your environment.

# **Custom Group Types and Fields for Grouper v1.2.1**

As shipped, Grouper groups have 6 attributes and 1 list, and every Grouper group has those 7 fields without exception (see the <u>Grouper Glossary</u> for a list of them and other Grouper-specific terms used in this section). Deployers can augment the pool of available fields with their own *Custom Fields*. These are gathered into sets of custom fields to form a *Group Type*, and all defined group types are available to be assigned to individual groups by their ADMINs. Assignment of a group type to a group adds the associated set of fields to that group. These custom fields can then be managed or accessed by the API or the UI, appear in XML exports, etc.

In this section we describe two methods for loading group types, i.e., collections of custom fields, into your Groups Registry, and one method for deleting them. Before that, however, you'll need to know a bit more about Grouper fields.

Fields come in two flavors: **attributes** and **lists**. Attributes have a single, string value. List fields are lists of subjects. Each field must have a declared **Access Privilege** necessary to read the field, and likewise an access privilege declared that's needed to write the field. And some fields are "required" - the required fields associated with a given group must all have a value, a requirement that is only enforced by an API caller (including the Grouper UI).

So, to create a custom group type is to declare its name, identify the names of fields associated with that type, define the type of each field (attribute or list), its read and write access privileges, and whether or not it is required. Described below are two means for so doing.

#### Using the XML Import tool to add a group type

The XML Import tool's metadata import capability can be used to load custom group types. Below is an example XML file that declares the group type named 'teaching' and its three associated fields, 'academic-staff', 'clerical-staff', and 'faculty\_code'. The first two of these are lists and the third is an attribute, and the privileges necessary to read or write them is also declared. None of these fields are in fact required.

```
<registry>
<metadata>
<groupTypesMetaData>
<groupTypeDef name='teaching'>
<field name='academic-staff' required='false' type='list' readPriv='read' writePriv='update'/
>
<field name='clerical-staff' required='false' type='list' readPriv='read writePriv='update'/>
<field name='faculty_code' required='false' type='attribute' readPriv='read'
writePriv='admin'/>
</groupTypeDef>
</metadata>
</registry>
```

To successfully import this XML into your Groups Registry, the import.properties file must include the following declarations:

```
import.metadata.group-types=true
import.metadata.group-type-attributes=true
```

Please refer to the XML Import/Export Tool documentation for details on how to load this XML.

#### Using GrouperShell to create a group type

The Grouper API's <u>GroupType method</u> can be used directly to create types and fields. Here's an example of using the GrouperShell to create the "teaching" group type presented in the XML above:

```
gsh-0.0.1 0% subj=SubjectFinder.findById("GrouperSystem")
subject: id='GrouperSystem' type='application' source='g:isa' name='GrouperSystem'
gsh-0.0.1 1% sess=GrouperSession.start(subj)
edu.internet2.middleware.grouper.GrouperSession: e161f71d-19f3-4cef-b6b8-
f0de9b594aab, 'GrouperSystem', 'application'
gsh-0.0.1 2% type=GroupType.createType(sess, "teaching")
edu.internet2.middleware.grouper.GroupType: teaching
gsh-0.0.1 3% read=Privilege.getInstance("read")
edu.internet2.middleware.grouper.Privilege: read
gsh-0.0.1 4% update=Privilege.getInstance("update")
edu.internet2.middleware.grouper.Privilege: update
gsh-0.0.1 5% admin=Privilege.getInstance("admin")
edu.internet2.middleware.grouper.Privilege: admin
gsh-0.0.1 6% type.addList(sess, "academic-staff", read, update)
edu.internet2.middleware.grouper.Field: academic-staff,teaching,list
gsh-0.0.1 7% type.addList(sess, "clerical-staff", read, update)
edu.internet2.middleware.grouper.Field: clerical-staff,teaching,list
gsh-0.0.1 8% type.addAttribute(sess, "faculty_code", read, admin, false)
edu.internet2.middleware.grouper.Field: faculty_code,teaching,attribute
```

#### Using GrouperShell to remove a group type

The Grouper API's GroupType and <u>GroupTypeFinder</u> methods can be used delete a group type. Here's an example of using the GrouperShell to delete the "teaching" group created above:

```
gsh-0.0.1 0% subj=SubjectFinder.findById("GrouperSystem")
subject: id='GrouperSystem' type='application' source='g:isa' name='GrouperSystem'
gsh-0.0.1 1% sess=GrouperSession.start(subj)
edu.internet2.middleware.grouper.GrouperSession: eb0c794b-09a1-4cc5-
b45b-4c2e4a6d3434,'GrouperSystem','application'
gsh-0.0.1 2% type=GroupTypeFinder.find("teaching")
edu.internet2.middleware.grouper.GroupType: teaching
gsh-0.0.1 3% type.delete(sess)
```

Questions or comments? <sup>1</sup> Contact us.

**<u>GROUPER</u>: About FAQ Software Documentation Archives Contribute WG Contact</u>** 

# **Customising the Grouper UI**

This page last changed on May 28, 2008 by tbarton@uchicago.edu.

**<u>GROUPER</u>: About FAQ Software Documentation Archives Contribute WG Contact</u>** 

# How to Customise the Grouper User Interface (UI)

This document is current as of the Grouper v1.3.0 release.

- Introduction
- <u>CSS Changes</u>
- <u>Changing the Internet2 logo</u>
- <u>Changing the Default Text</u>
- <u>Using Custom Templates Instead of the Standard Templates</u>
- Defining Custom Dynamic Templates
- Modifying Existing Struts Actions, Adding New Actions, and Making New Tiles Definitions Available
- <u>Customising authentication</u>
- <u>Customising group authorizations</u>
- <u>Customising the Root Node of the Grouper Repository</u>
- <u>Creating an InitialStems View</u>
- <u>Customising Browsing and Searching</u>
- Customising the Menu
- Personal Groups
- Displaying subjects, groups and stems
- Sort order of lists
- Enabling import / export of group memberships
- <u>Customising the Build Process</u>
- <u>Customising web.xml</u>
- Running the Standard UI at the Same Time as the Custom UI
- Determining How a Grouper UI Page Was Constructed
- Providing Feedback and Getting Help

### Introduction

This document is written for the web application developer. It describes how to make changes to the Grouper UI and is best understood with reference to the Grouper UI architecture (which contains links to underlying concepts and technologies). The section <u>Determining How a Grouper UI page Was Constructed</u> explains how to display on screen information, which can help determine what you need to change in order to modify a Grouper UI page. <u>Struts Actions</u> in the Grouper UI gives an overview of which Struts actions relate to which functional areas.

The document focuses on the specific customisations which can be built into the QuickStart demo (see the QuickStart README), and will refer to differences between the standard and custom screen shots in <u>Grouper UIs</u>. You may want to open this link in a separate window / tab for reference.

The UI has been designed so that the source code for the standard UI need not be changed in order to effect customisations. This is intended to make it easier to upgrade changes to the standard UI without compromising customisations you have made.

The structure and contents of **grouper-qs/custom-grouper-ui** should be used as the starting point for your own custom Grouper UI.

### **CSS Changes**

Grouper has its own CSS stylesheets, but provides a mechanism for site-specific stylesheets to be loaded after the Grouper stylesheets. This allows sites to override/extend existing styles and to add new styles. Such changes can alter the position of screen elements, fonts, colours, etc.

The custom stylesheet was configured in **grouper-qs/custom-grouper-ui/resources/custom/ media.properties**:

#### css.additional=custom/custom.css

The actual stylesheet is found at grouper-qs/custom-grouper-ui/webapp/custom/custom.css.

**Note**: As of version 0.9 of Grouper css.additional can be a space separated list of stylesheets. It is also possible to 'turn off' the Grouper stylesheets by setting the key grouper-css.hide=true.

Differences in the standard and custom UIs which are due to CSS are listed below:

Feature	Standard UI	Custom UI
Menu	Positioned vertically on left.	Positioned horizontally below logos. Various colour changes.
Content area	To the right of the menu.	Aligned left and full width.
'Members'	Blue text on pale background.	White text on grey background.

Many more CSS classes are applied to elements in the HTML source than are specified in the Grouper stylesheets so a high degree of CSS customisation is possible.

It is possible to turn off the style sheets. This may be useful for testing the accessibility of the Grouper UI and any customisations you make (see the *Remove CSS stylesheet references?* form field in <u>Determining</u> <u>How a Grouper UI Page Was Constructed</u>.

### **Changing the Internet2 Logo**

In grouper-qs/custom-grouper-ui/resources/custom/media.properties, the following property was set:

#### image.organisation-logo=custom/images/banner.logo.gif

## **Changing the Default Text**

In the standard UI, all navigational text and instructions are derived from a Java ResourceBundle based on **grouper-qs/grouper-ui/resources/grouper/nav.properties**.

In the custom UI, values for keys set in **grouper-qs/custom-grouper-ui/resources/custom/ nav.properties** will override the standard UI values. In the UI example screen shots, the custom UI includes *UoB* in menu items and changes *Help* to*Assistance*.

Through the use of Java ResourceBundles, the Grouper UI supports Internationalization. The default locale for the standard UI is set in **grouper-qs/grouper-ui/resources/init.properties**:

default.locale=en\_US

In **grouper-qs/custom-grouper-ui/resources/init.properties,default.locale=en\_GB**, however, there is no *nav\_en\_GB.properties* file so there are no differences due to locale in the standard and custom UIs.

Currently, there is nowhere in the UI to select a different locale from the default, however, if a *lang* parameter is passed as part of the URL which invokes login, the value will be used as the locale for the current session.

See <u>Determining How a Grouper UI Page Was Constructed</u> for details of how to display which key / value pairs were used in an actual page in the UI.

If you create your own templates you are not under any obligation to use ResourceBundles instead of directly entering text in templates, however, if you wish to contribute code back to Grouper, such a contribution would be more useful if it used ResourceBundles.

### **Using Custom Templates Instead of the Standard Templates**

The Grouper UI uses Strut's Tiles to define core page components. In the standard UI these are defined in **grouper-qs/grouper-ui/webapp/WEB-INF/tiles-def.xml**. In the custom UI some definitions are modified and another added in the file**grouper-qs/custom-grouper-ui/webapp/WEB-INF/tilesdef-custom.xml**. New definitions for *headerDef* and *footerDef* allow site specific branding. *groupStuffDef* defines a template which is included in the Group Summary page of the UI.

*EasyLoginFormDef* defines a new page (see <u>Customising Authentication</u>).

### **Defining Custom Dynamic Templates**

Grouper recognises some core entities such as Groups, Stems, Subjects and Collections. The Grouper UI dynamically chooses the appropriate template for an entity at runtime based on its type and the UI context. The default templates for an entity and view are defined in **grouper-qs/grouper-ui/ resources/grouper/media.properties**. When a specific entity-view key is not found, a default is used. Key-value pairs can be overridden, or more specific keys added to **grouper-qs/custom-grouper-ui/ resources/grouper/media.properties**. The algorithms used to choose appropriate keys are described in the Javadoc for <u>DefaultTemplateResolverImpl</u>. This class can be extended to add support for other entity types, or a completely new implementation plugged in (see Javadoc for the <u>TemplateResolver</u> interface.

In the standard UI the default template for a subject is defined:

#### subject.view.default=/WEB-INF/jsp/subjectView.jsp

In the custom UI:

#### personQS.view.default=/WEB-INF/jsp/custom/customPersonSubjectView.jsp

defines a specific template for subjects whose source has an ID of personQS. In the UI examples the name Keith Benson is followed by (kebe) in the custom UI due to the use of **/WEB-INF/jsp/custom/customPersonSubjectView.jsp** as a template. Subjects and groups may have any number of site specific attributes, so dynamic templates allow sites to create templates which have access to these custom attributes.

See <u>Determining How a Grouper UI Page is Constructed</u> for determining which template was chosen for an entity-view on a page in the UI.

### Modifying Existing Structs Ations, Adding new Actions, and Making New Tiles Definitions Available

The Grouper UI is based on Struts and the standard Struts configuration is through **grouper-qs/ grouper-ui/webapp/WEB-INF/struts-config.xml**. Existing actions can be replaced and new ones added to **grouper-qs/custom-grouper-ui/webapp/WEB-INF/struts-config-custom.xml**.

See <u>Struts Actions in the Grouper UI</u> for an explanation of how the standard Grouper ui actions interact.

In the custom UI the action /callLogin is redefined such that it forwards to /easyLogin - a new action.

More information on how to change the behaviour of the Grouper Struts actions is available in the appropriate <u>javadoc package description</u>.

Even if you don't need to change/add any actions, a Tiles plugin must be configured in order to make custom templates available (see <u>Using Custom Templates</u> instead of the standard templates):

```
<plug-in className="org.apache.struts.tiles.TilesPlugin">
    <set-property property="moduleAware" value="true"/>
    <set-property property="definitions-debug" value="0"/>
    <set-property property="definitions-parser-details" value="0"/>
    <set-property property="definitions-parser-validate" value="false"/>
```

```
<set-property property="definitions-config" value="/WEB-INF/tiles-def.xml,/WEB-INF/tiles-def-
custom.xml"/>
</plug-in>
```

Note that the order of files in the definitions-config property is important as the last Tile definition with a particular name loaded is used.

### **Customising Authentication**

The standard UI uses basic HTTP authentication configured through Tomcat and the web application web.xml file. A Filter LoginCheckFilter checks if you are logged in before allowing access to the application. It checks the **javax.servlet.http.HttpServletRequest.getRemoteUser()**. If not set the user is redirected to the splash page, otherwise, access is granted, and if necessary, the user session initialised.

In the custom UI, when a user clicks the *Login* link on the splash page, the */callLogin* action is requested. This forwards the user to the */easyLogin* action which displays the template named *EasyLoginFormDef*, which is a simple form allowing a username to be entered. The custom UI also defines a Filter *EasyLoginFilter* which is called prior to LoginCheckFilter. If it *sees* a request parameter called username, it attempts to load a Subject (through *SubjectFinder.findByIdentifier*). If successful, it stores the username in the HttpSession and calls the next Filter in sequence with a modified *HttpServletRequest*, which responds to getRemoteUser() by returning the stored username.

This section shows how new authentication schemes can be introduced. A more serious scheme that allows <u>Yale CAS Authentication</u> has been contributed to the Grouper UI distribution.

In order to configure new Filters, it is necessary to modify the web application web.xml file. How to do this is described in <u>Customising web.xml</u>.

Note: The standard UI does not have a logout link, because it is not possible to safely logout of basic HTTP authentication. Other authentication schemes will generally work by setting an HttpSession attribute - which is cleared when an HttpSession is invalidated, so a logout link is provided.

### **Customising group authorizations**

As of v1.3.0 it is possible to override how the UI decides what the current user can do with a group. The primary motivation for this feature is to allow some UI features to be turned off i.e. when the feature should be maintained by a loader. Customization is achieved by implementing the <u>UIGroupPrivilegeResolver</u> interface, and configuring it through media.properties e.g.

edu.internet2.middleware.grouper.ui.UIGroupPrivilegeResolver=uk.ac.bris.is.grouper.ui.UoBUIGroupPrivilegeResolver=uk.ac.bris.grouper.ui.UoBUIGroupPrivilegeResolver=uk.ac.bris.grouper.ui.UoBUIGroupPrivilegeResolver=uk.ac.bris.grouper.ui.UoBUIGroupPrivilegeResolver=uk.ac.bris.grouper.ui.UoBUIGroupPrivilegeResolver=uk.ac.bris.grouper.ui.UoBUIGroupPrivilegeResolver=uk.ac.bris.grouper.ui.UoBUIGroupPrivilegeResolver=uk.ac.bris.grouper.ui.ui.UoBUIGroupPrivilegeResolver=uk.ac.bris.grouper.ui.ac.bris.grouper.ui.ui.UoBUIGroupPrivilegeResolve

### **Customising the Root Node of the Grouper Repository**

#### \*see also Customizing Browsing and Searching

The Grouper API defines a root stem. In the standard UI the **Current location** begins with *Root* whereas in the custom UI it starts with *QS University of Bristol*. Both screen shots are views of the same Grouper repository. Three scenarios are possible

- 1. Root is visible, and browsing starts at Root,
- 2. Root is visible but browsing starts at a defined stem e.g. *QS University of Bristol.*
- 3. Root is not visible and browsing starts at a defined stem e.g.QS University of Bristol.

As of Grouper 0.9 it is possible to specify a root node per browse mode (see <u>AbstractRepositoryBrowser</u>). If none is specified, **default.browse.stem** from media.properties is used. If this is not set the the root node is used and scenario 1 occurs. If the root node is displayed, its name is determined by **stem.root.display-name** from nav.properties. If this is not set 'Root is used by default.

By default, the hide-pre-root-node value for each *RepositoryBrowser* defined in media.properties, is set to *true*. This leads to scenario 3.

If hide-pre-root-node is set to false then scenario 2 occurs.

### Creating an *InitialStems* View

\*see also Customizing Browsing and Searching

This feature is not implemented in either the standard or custom UIs; however, it provides an alternative starting point for browsing, by allowing sites to provide a customised list of stems or *quick links*. The list of stems can come from any part of the hierarchy, and so may provide a better starting point for users, i.e., for GrouperSystem the default view is:

- Personal Groups
- Academic Faculties
- Student Union
- Non-Academic Departments
- Community Groups
- [All Students]
- [All Academic Staff]
- [All Students and Academic Staff]
- [UoB Administrators]

But for another user (e.g., an art student), the following list might be more appropriate:

- Personal Groups for <name>
- Arts Faculty
- Student Union Clubs

Such a list could be generated in a site-specific way based on a username. A site might also provide a means for a user to edit their list of quick links.

See Javadoc for InitialStems interface.

As of Grouper 0.9 it is possible to define a different InitialStems implementation for each browse mode; see <u>AbstractRepositoryBrowser.getInitialStems()</u>

### **Customising Browsing and Searching**

As of version 0.9 of Grouper it is possible to customise existing browse modes and add new browse modes. It is also possible to specify *root nodes* and *InitialStem* implementations on a mode by mode basis.

At runtime <u>RepositoryBrowserFactory</u> is used to obtain a <u>RepositoryBrowser</u> implementation for the current browse mode, by obtaining the class name of the implentation from **resources/grouper/ media.properties** using the key **repository.browser.<mode>.class**. All Grouper supplied implementations extend <u>AbstractRepositoryBrowser</u>, which reads further properties. Thus, the behaviour of supplied browse modes can be modified by changing relevant properties. The logic can be modified by providing a new implementation class - possibly a subclass of the Grouper implementation.

Alternatively, new browse modes can be implemented and configured. In general you will also need to implement a top level Strut's action and page for any new browse mode, and provide links as appropriate. See <u>Customising the Menu</u> for details on how to change the default menu.

### **Customising the Menu**

As of version 0.9 of Grouper the menu is configurable. <u>PrepareMenuAction</u>reads **resources/grouper/ menu-items.xml** to obtain a list of known menu items. A **media.properties** key, **menu.order**, determines the order in which items are rendered. Sites can add additional menu items by creating their own menu-items.xml and adding the file name to the **media.properties** key: **menu.resource.files**.

If sites want to have different menus for different users they can subclass **PrepareMenuAction** and override the **protected boolean isValidMenuItem(Map item,GrouperSession grouperSession,HttpServletRequest request)** method. You will also need to override the Strut's action **prepareMenu.do**.

As of version 1.2.1 a new <u>MenuFilter</u> interface has been introduced to allow a more structured approach to customizing menus. Two implementations are provided, configured as:

menu.filters=edu.internet2.middleware.grouper.ui.RootMenuFilter
 edu.internet2.middleware.grouper.ui.GroupMembershipMenuFilter

<u>GroupMembershipMenuFilter</u> is configured using a <u>UiPermissions</u> object and allows menu items to be vetoed depending on whether or not a user is a member of a group.

#### **Personal Groups**

Grouper has no specific support for personal groups, however, by implementing the <u>PersonalStem</u> interface, the Grouper UI will create a 'personal stem' for a user (if one does not exist) at login. An implementation of *PersonalStem* is provided at **grouper-qs/custom-grouper-ui/java/src/edu/ internet2/middleware/grouper/customqs/ui/CustomQSPresonalStem.java**. This implementation creates a stem (extension=subject id) as a child of **/qsuob/personal**. Currently any user who is logged in can see personal stems. Whether they can see groups in the personal stem will depend upon Access privileges. Sites could use custom implementations of RepositoryBrowsers to implement their own business rules around personal stems and groups.

#### Displaying subjects, groups, and stems

Prior to Grouper v1.2.0 it was necessary to use custom dynamic tiles to change how subjects, groups and stems are displayed. This still remains the most flexible approach, especially if you need to show more than one attribute.

As of Grouper v1.2.0 it is possible to configure a single arbitrary attribute to use when displaying a subject, group or stem. The default media.properties keys are:

#Default if an attribute is not configured for a specific subject source. 'description' is set for backwards compatability subject.display.default=description

#used for subjects which are groups sourced by Grouper subject.display.g\:gsa=displayExtension

#used for internal subjects i.e. GrouperSystem and GrouperAll subject.display.g\:isa=name

#default attribute for groups (when not viewed as a subject)
group.display=displayExtension

#flat = context i.e. flat mode in the UI. Here the hierarchy is not shown and names displayExtension need not be unique across #multiple stems. group.display.flat=displayName

#default attribute for stems
stem.display=displayExtension

#### In the QuickStart the following key is also used:

subject.display.qsuob=name

When displaying search results sites can configure a default attribute to display:

search.group.result-field=name
search.stem.result-field=name

in addition sites can configure a set of attributes from which the user may select one to display:

search.group.result-field-choice=name displayExtension displayName
search.stem.result-field-choice=name displayExtension displayName

# As of Grouper v1.2.1 it is possible, for the SubjectSummary page, to specify a subset of available attributes to display and the order in which to display them:

# subject.attributes.order.<SOURCE\_ID>=comma separated list of case sensitive attribute names subject.attributes.order.g

 $\verb|:gsa=displayExtension,displayName,name,extension,createTime,createSubjectId,createSubjectType,modifySubjectId,createSubjectType,modifySubjectId,createSubjectType,modifySubjectId,createSubjectType,modifySubjectId,createSubjectType,modifySubjectId,createSubjectType,modifySubjectId,createSubjectType,modifySubjectId,createSubjectType,modifySubjectId,createSubjectSubjectType,modifySubjectId,createSubjectSubjectSubjectType,modifySubjectId,createSubjectSub$ 

#subject.attributes.order.qsuob=LOGINID,LFNAME,subjectType,id

The UI wraps API objects as specific subclasses of <u>ObjectAsMap</u>. As of v1.3.0 it is possible to configure your own implementations. Typically these would subclass the Grouper UI concrete subclasses and add/ modify behaviour. Configuration is through media.properties e.g.

objectasmap.StemAsMap.impl=uk.ac.bris.is.grouper.ui.util.UOBStemAsMap

You could use this feature to create virtual attributes as composites of other attributes.

### Sort order of lists

As of Grouper v1.2.0 various lists may be sorted alphabetically.

- search results for:
  - ° groups
  - ° stems
  - subjects
- group memberships
- group privilegees
- stem privilegees
- · groups / stems where a subject has a selected privilege
- saved subjects
- saved groups
- stems / groups whilst browsing
- stems / groups in *flat* mode

See Javadoc for <u>LowLevelGrouperCapableAction.html.sort</u> and <u>DefaultComparatorImpl</u> for detailed information.

In principle, the sort algorithm should use exactly what is displayed on screen to sort lists, and, by default, this is what it does. The sort algorithm, therefore, takes account of the configuration for <u>Displaying subjects</u>, <u>groups and stems</u>. On the other hand, Grouper allows the use of dynamic tiles and so it is possible to override the defaults in a way that the sort algorithm cannot work out. If a site does use dynamic tiles to display subjects, groups or stems, it is possible to configure Grouper to use alternate configuration for sorting, but it is the responsibility of the administrator to ensure that the sort configuration is appropriate for what is displayed on screen. For maximum flexibility, it is also possible to configure different attribute(s) to sort on for different contexts\*. The 'search' order for keys is documented for each implementation of <u>GrouperComparatorHelper</u>.

\*The different contexts recognised are:

- search
- members
- flat
- subjectSummary
- privilegees

As of Grouper v1.2.1 you can sort subjects from the same source together by defining strings which are pre-pended to the usual sort string:

## Enabling import / export of group memberships

As of Grouper v1.2.0 it is possible to configure the UI to enable import / export of group memberships. Simple implementation classes are provided for dealing with tab or comma delimited files. In general, the formats for import or export vary for different sites. By default import / export is not enabled. Import is controlled by <u>MembershipImportManager</u> and a <u>DefaultMembershipImporter</u> class is provided. Export is controlled by <u>MembershipExporter</u>.

In the QuickStart import and export is 'activated' using:

```
membership-export.config=resources/custom/membership-export.xml
membership-import.config=resources/custom/membership-import.xml
```

You can adapt these configuration files for your own needs and even write your own import implementation if the classes provided are unsuitable.

As of Grouper v1.2.1 you can configure the UI to allow import of data from a text area:

membership-import.allow-textarea=true

If a user does not select a file to import, the user is presented with a text area where they can type or paste data.

### **Customising the Build Process**

The Grouper UI uses the **grouper-qs/grouper-ui/build.xml** ant script to build the web application. This script is configured through**grouper-qs/grouper-ui/build.properties**, which has a key **additional.build**. In the custom UI this is set to **\\${basedir}/**../custom-grouper-ui/additionalbuild.xml. It is the responsibility of this script, which is called by the standard script, to compile any Java source files and to copy to the build area any other necessary files. If you wish to incorporate any contributed code, calls to the relevant build scripts should be placed here. In the **custom-grouper-ui/ additional-build.xml** script, the struts-patch build script is called.

### **Customising web.xml**

A web application web.xml file is a key configuration file and any site wishing to customise the Grouper UI will need to modify it. The web.xml is a J2EE deployment descriptor which configures the Servlets (how URLs are mapped to Java classes), the filters (pre/post logic around servlets), j2ee security (if not done in apache or somewhere else), listeners (for j2ee events), custom tag libraries (how some tags in JSPs map to java classes), etc. Things you might need to customize are filters (e.g. a new way to do authentication / authorization), security (do you want the servlet container to manage authentication / authorization?), custom tag libraries (are you using a new library in JSP extensions?), etc.

The default deployment descriptor is found at **grouper-qs/grouper-ui/webapp/WEB-INF/ web.core.xml**. The UI provies a mechanism for merging fragments of different web.xml files into a final deployment file. In your additional build script (see <u>Customising the Build Process</u> copy any web.xml fragments into **\\${temp.dir}**. Typically files should be prefixed with a 2 digit number e.g. 20 (90 is used for web.core.ml). The merging process merges in name order of the files. The custom UI includes two web.xml fragments which, when copied, are prefixed with 00 and 95 so the former is merged with web.core.xml and the latter is merged with the result of the first merge.

The first web.xml fragment is **grouper-qs/custom-grouper-ui/webapp/WEB-INF/web.custom.xml** and it overrides the default action servlet definition to ensure that it loads the Struts config customisations - which in turn load the Tiles definition customisations.

<servlet>

<sup>&</sup>lt;servlet-name>action</servlet-name>

<sup>&</sup>lt;servlet-class>org.apache.struts.action.ActionServlet</servlet-class>

<sup>&</sup>lt;init-param>

```
<param-name>config</param-name>
  <param-value>/WEB-INF/struts-config-custom.xml,/WEB-INF/struts-config.xml,/WEB-INF/struts-
config-custom.xml</param-value>
  </init-param>
  <init-param>
   <param-name>config/i2mi</param-name>
   <param-name>config/i2mi</param-name>
   <param-value>/WEB-INF/struts-config.xml</param-value>
  </init-param>
   <param-value>/WEB-INF/struts-config.xml</param-value>
  </init-param>
   <param-value>/WEB-INF/struts-config.xml</param-value>
  </init-param>
   <param-value>/WEB-INF/struts-config.xml</param-value>
  </init-param>
```

Note: As the order of elements in the final web.xml file is important it can be difficult to get elements in the order you want. The merging process is not extensively tested and it is quite likely it will not work properly for all elements. It may be necessary to rework the merging process, or resort to manual editing of the web.core.xml file.

The merge process is dependent on **web-xml-merge.xsl**and **web-xml-merge-tags.xml** found in **grouper-qs/grouper-ui**.

### Running the standard UI at the same time as the custom UI

By applying the <u>struts-patch</u> contribution (see <u>Customising the Build Process</u>), and configuring the Struts action servlet with more than one module see (**config/i2mi** parameter in <u>Customising web.xml</u>), it is possible to have both the standard and custom UIs available at the same time.

After building the custom Grouper UI you appear to *lose* the standard UI, however, if instead of accessing */grouper*, you access */grouper/i2mi* in your web browser, you then get the standard UI.

### **Determining How a Grouper UI Page Was Constructed**

Since a Grouper UI page may be constructed from many templates, including dynamic templates, and it may not be easy to determine which ResourceBundle key was used to render text, a mechanism has been created to display *debug* information. Go to the URI /grouper/populateDebugPrefs.do. You should see a form:

The form fields are explained in the table below:

Field	Description
Enable debug display	Determines if debug information is shown at the bottom of the page. If not selected, none of the other options are active.
Webapp root for I2mi*	The complete file system path to the standard UI webapp root.
Webapp root for your site*	The complete file system path to the custom UI webapp root.
Show resource keys and values at end of page	If selected, any key-value pairs derived from the nav ResourceBundle to display screen text are listed.
Show resource keys in page rather than values	Instead of seeing the text in the page you will see ?key?
Show dynamic tiles	If selected, a hierarchy of templates used to construct the page is shown. If a template was loaded dynamically, the <i>view, entity type, chosen key</i> and <i>template name</i> are displayed.

#### Executable for JSP editor

The complete filesystem path to a JSP editor - only use if the server is your working machine! If the webapp roots above are specified, template names are links which will open the template file in the specified JSP editor. Remove CSS stylesheet references?

Allows you see the non stylised page - used for checking accessibility in absence of a screen reader.

\*These fields are only required if you wish to link template names to a JSP editor.

# **Providing Feedback and Getting Help**

The Grouper UI is intended to be extensible and not to force unnecessary constraints, however, it is only as sites try to make their own customisations that the true extensibility can be tested. If while customising the Grouper UI you find yourself forced to modify standard Grouper UI sources (of any kind), or find that you cannot easily do what you want to, please offer feedback to, or request help via the grouper-users mailing list.

Questions or comments? <sup>1</sup> Contact us.

GROUPER: About FAQ Software Documentation Archives Contribute WG Contact

# **Grouper UIs**



This page last changed on May 28, 2008 by tbarton@uchicago.edu.

# **Developer's Guide to the Grouper API**

This page last changed on Dec 06, 2007 by jbibbee@internet2.edu.

**<u>GROUPER</u>:** <u>About</u> <u>FAQ</u> <u>Software</u> <u>Documentation</u> <u>Archives</u> <u>Contribute</u> <u>WG</u> <u>Contact</u>

#### Using the Grouper API to bootstrap your Groups Registry

This document is current as of the v1.2.0 release.

- Find GrouperSystem Subject (example code)
- Start-and-stop sessions (example code)
- Find the root stem (example code)
- Find stem by name(example code)
- Create stem (example code)
- Find group by name (example code)
- <u>Create group</u> (example code)
- Find GrouperAll subject (example code)
- <u>Check for membership in the wheel group (example code)</u>
- Add wheel group member (example code)

**Questions or comments? Contact us.** 

**<u>GROUPER</u>: About FAQ Software Documentation Archives Contribute WG Contact</u>** 

# **Getting Started**

This page last changed on Jan 04, 2009 by tbarton@uchicago.edu.

#### **<u>GROUPER</u>:** <u>About</u> <u>FAQ</u> <u>Software</u> <u>Documentation</u> <u>Archives</u> <u>Contribute</u> <u>WG</u> <u>Contact</u>

# Getting started with the Grouper API Binary Distribution v1.4.0

This page contains instructions for installing and configuring a new Grouper v1.4.0 installation.

- Meet required prerequisites and download the Grouper API binary release
- <u>API Building & Configuration</u>. Config files are in the conf/ directory. Config files ending in ".example" are templates and should not be modified. Config files not ending in ".example" are customizable
- Initialize your database (by default HSQLDB) by running gsh -registry -check -runscript

```
C:\dev_inst\eclipse\workspace_v33\grouper\bin>gsh -registry -check -runscript
Using GROUPER_HOME: C:\dev_inst\eclipse\workspace_v33\grouper\bin\..
Using GROUPER_CONF: C:\dev_inst\eclipse\workspace_v33\grouper\bin\../conf
Using JAVA:
                    "c:\dev_inst\java/bin/java"
                    64m-512m
using MEMORY:
Grouper starting up: version: 1.4.0 build date: 2008/11/13 14:42:25
grouper.properties read from: C:\dev_inst\eclipse\workspace_v33\grouper\conf\grouper.properties
Grouper current directory is: C:\dev_inst\eclipse\workspace_v33\grouper\bin
log4j.properties read from: C:\dev_inst\eclipse\workspace_v33\grouper\conf\log4j.properties
Grouper is logging to file: C:\dev_inst\eclipse\workspace_v33\grouper\bin\..\logs
\grouper_error.log, at min level WARN
for package: edu.internet2.middleware.grouper, based on log4j.properties
grouper.hibernate.properties: C:\dev_inst\eclipse\workspace_v33\grouper\conf
\grouper.hibernate.properties
grouper.hibernate.properties: sa@jdbc:hsqldb:C:\dev_inst\eclipse\workspace_v33\grouper\bin\..\dist/
run/grouper;create=true
sources.xml read from:
                             C:\dev_inst\eclipse\workspace_v33\grouper\conf\sources.xml
sources.xml groupersource id: g:gsa
sources.xml jdbc source id: jdbc: GrouperJdbcConnectionProvider
Based on grouper.properties: ddlutils.schemaexport.installGrouperData=true
(note, might need to type in your response multiple times (Java stdin is flaky))
(note, you can whitelist or blacklist db urls and users in the grouper.properties)
Are you sure you want to schemaexport all tables (dropThenCreate=F,writeAndRunScript=T) in
db user 'sa', db url 'jdbc:hsqldb:C:\dev_inst\eclipse\workspace_v33\grouper\bin\..\dist/run/
grouper;create=true'? (y|n):
У
Continuing...
Grouper ddl object type 'Grouper' has dbVersion: 0 and java version: 12
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:
C:\dev_inst\eclipse\workspace_v33\grouper\bin\..\ddlScripts\grouperDdl_20081114_02_00_20_771.sql
Script was executed successfully
```

C:\dev\_inst\eclipse\workspace\_v33\grouper\bin>

- Verify the installation was successful by examining log files in the logs/ directory.
- When using HSQLDB, a second attempt is sometimes necessary: gsh -registry -check -runscript
- It is now possible to run GrouperShell (gsh) and Initialize Administration of Privileges

C:\dev\_inst\eclipse\workspace\_v33\grouper\bin>gsh Using GROUPER\_HOME: C:\dev\_inst\eclipse\workspace\_v33\grouper\bin\.. Using GROUPER\_CONF: C:\dev\_inst\eclipse\workspace\_v33\grouper\bin\../conf Using JAVA: "c:\dev\_inst\java/bin/java" using MEMORY: 64m-512m Grouper starting up: version: 1.4.0 build date: 2008/11/13 14:42:25 grouper.properties read from: C:\dev\_inst\eclipse\workspace\_v33\grouper\conf\grouper.properties Grouper current directory is: C:\dev\_inst\eclipse\workspace\_v33\grouper\bin log4j.properties read from: C:\dev\_inst\eclipse\workspace\_v33\grouper\conf\log4j.properties Grouper is logging to file: console, at min level WARN for package: edu.internet2.middleware.grouper, based on log4j.properties grouper.hibernate.properties: C:\dev\_inst\eclipse\workspace\_v33\grouper\conf \grouper.hibernate.properties grouper.hibernate.properties: sa@jdbc:hsqldb:C:\dev\_inst\eclipse\workspace\_v33\grouper\bin\..\dist/ run/grouper;create=true sources.xml read from: C:\dev\_inst\eclipse\workspace\_v33\grouper\conf\sources.xml sources.xml groupersource id: g:gsa sources.xml jdbc source id: jdbc: GrouperJdbcConnectionProvider 2008-11-14 02:12:06,041: [main] WARN ApiConfig.printConfigOnce(189) - Grouper starting up: version: 1.4.0 build date: 2008/11/13 14:42:25 Type help() for instructions gsh 0% addRootStem("penn", "penn"); stem: name='penn' displayName='penn' uuid='b693f8f1-9763-44ae-a984-be25e0b6ea0f' gsh 1% addGroup("penn", "faculty", "faculty"); group: name='penn:faculty' displayName='penn:faculty' uuid='fbf3646f-eb10-47c1-9cc0-d32b7049a336' gsh 2%

GROUPER: About FAQ Software Documentation Archives Contribute WG Contact
### Glossary

This page last changed on May 16, 2008 by bmk@isc.upenn.edu.

**<u>GROUPER</u>:** <u>About</u> <u>FAQ</u> <u>Software</u> <u>Documentation</u> <u>Archives</u> <u>Contribute</u> <u>WG</u> <u>Contact</u>

# **Grouper Glossary**

Terms with Grouper-specific meaning are defined below, along with other Grouper concepts. An understanding of these terms will enable you to take full advantage of all that Grouper has to offer.

As of v1.3.0, <u>terminology used in the Grouper UI</u> differs from some of the terms defined below to help the UI to present group management tasks in a manner more readily understandable by non-technical users. The terminology used in developer and system administrator oriented documentation remains unchanged.

**Note:** To view an HTML version of this document, open the <u>attachment</u> above.

TERM	DEFINITION	UI Translation (where applicable)
Access Privileges	Privileges that determine what a <b>Subject</b> can do with a <b>Group</b> . They are:	Subject is a UI "entity"
Attribute	<ul> <li>ADMIN - can assign access privileges and manage all group information,</li> <li>UPDATE - can manage membership of the group (implies READ),</li> <li>READ - can see the membership of the group (implies VIEW), and</li> <li>VIEW - can see the group. In addition, a group may have options for its members to:</li> <li>OPTIN - can add self to the membership, and</li> <li>OPTOUT - can remove self from membership.</li> <li>A single-valued string associated with a <i>Group</i> or a <i>Naming Stem</i>. By default, Grouper supports six attributes (one of two kinds of <i>Field</i>):</li> <li>id - a Grouper-assigned, globally unique identifier.</li> <li>extension - the relative name of the group or naming stem; the contribution of a single element, such as a group or a naming stem, to the cumulative name.</li> <li>name - used to facilitate searching for groups by name, it is a read-only string representation of the logical ordered pair of (<i>parent stem</i>, <i>extension</i>). This attribute is system-maintained. The string representation of the</li> </ul>	<ul> <li>id is the UI "UUID"</li> <li>extension is the UI "ID"</li> <li>name is the UI "ID path"</li> <li>displayExtension is the UI "name"</li> <li>displayName is the UI "path"</li> </ul>

[	
	name attribute is: <pre>content</pre>
	steril>: <extension></extension>
	• displayExtension - d
	extension
	<ul> <li>displayName- used to</li> </ul>
	facilitate searching for
	arouns by the displayed
	name it is a read-only
	string representation of
	the logical ordered pair of
	(displayName of parent
	stem, displayExtension).
	This attribute is system-
	maintained. The string
	representation of the
	displayName attribute is:
	<displayname of="" parent<="" th=""></displayname>
	stem>: <displayextension>.</displayextension>
	description - a description
	of the group or naming stem.
Composite Group	A <i>Group</i> whose <i>Membership</i>
	is determined by combining
	the membership lists of two
	other groups, without listing
	its members explicitly. These
	two groups are called its <b>Factor</b>
	Groups. Three methods of
	combining the factor groups'
	memberships are supported:
	• union - all subjects must
	be a member of one OR the
	other factor group,
	e.q., Group $Z =$ members of
	either Group X <b>OR</b> Group Y,
	or $Z = X \cup Y$ .
	<ul> <li>intersection - all subjects</li> </ul>
	that are members of the first
	factor group AND the second
	factor group,
	e.g., Group $Z =$ members of
	both Group X <b>AND</b> Group Y,
	or $\angle = X \# Y$ .
	relative complement - all     mombars of the first factor
	aroun that are NOT members
	of the second factor aroun
	$e_{a}$ Group $7 = $ members of
	Group X <b>AND NOT</b> Group Y or
	7 = X - Y.
Direct Membershin	A <b>Subject</b> that is listed in the Subject is a UI "entity"
	Membership list of a Group has a
	direct membership in the group.
	Also see <b>Indirect Membership</b> .
Factor Group	A Group in combination (union,
•	intersection, or relative
	complement) with that of
	another factor group, which
	defines the membership of a
	resulting <b>Composite Group</b> .
Field	Lither an <b>Attribute</b> or a <b>List</b> .
	Grouper groups are a collection
	jot attributes and lists, i.e., a

	collection of fields. The set of fields
	attached to a given group is a
	function of the set of <b>Group Types</b>
	it has been assigned
	IL Has been assigned.
Group	A list of <b>Subjects</b> having fragether
	<b>Membership</b> In the group, together
	with other attributes about the
	group. A list can have zero or
	more entries. In Grouper, a list
	contains only subject references,
	and an attribute is a single-valued
	string. A group must be created
	in an existing <b>Naming Stem</b> . If a
	group is made a member, i.e., a
	Subaroup, of another group, the
	members of the group will also
	be made members. By default, a
	Grouper group has:
	• six naming <b>Attributes</b> ,
	• a <b>description</b> attribute, and
	• a members list
	This information model
	can be extended to include
	additional site defined
	adultional site-defined
	Any combination of groups for
Group Math	the number of groups for
	the purpose of creating another
	group based on the memberships
	of those groups. See <i>Composite</i>
	Group.
	A <b>Subject</b> that is a member of
Indirect Membership	A <b>Subject</b> that is a member of
Indirect Membership	a <i>Subgroup</i> of a <i>Group</i> , or a
Indirect Membership	a <i>Subgroup</i> of a <i>Group</i> , or a member of a <i>Factor Group</i> that
Indirect Membership	a <i>Subgroup</i> of a <i>Group</i> , or a member of a <i>Factor Group</i> that contributes positively to a group's
Indirect Membership	a <b>Subgroup</b> of a <b>Group</b> , or a member of a <b>Factor Group</b> that contributes positively to a group's membership, has an indirect
Indirect Membership	a <b>Subgroup</b> of a <b>Group</b> , or a member of a <b>Factor Group</b> that contributes positively to a group's membership, has an indirect membership in the group. Also see
Indirect Membership	a <b>Subgroup</b> of a <b>Group</b> , or a member of a <b>Factor Group</b> that contributes positively to a group's membership, has an indirect membership in the group. Also see <b>Direct Membership</b> .
List	a <b>Subgroup</b> of a <b>Group</b> , or a member of a <b>Factor Group</b> that contributes positively to a group's membership, has an indirect membership in the group. Also see <b>Direct Membership</b> . A multi-valued list of <b>Subject</b>
List	a <b>Subgroup</b> of a <b>Group</b> , or a member of a <b>Factor Group</b> that contributes positively to a group's membership, has an indirect membership in the group. Also see <b>Direct Membership</b> . A multi-valued list of <b>Subject</b> references, (one of two kinds of
List	<ul> <li>a Subgroup of a Group, or a</li> <li>member of a Factor Group that</li> <li>contributes positively to a group's</li> <li>membership, has an indirect</li> <li>membership in the group. Also see</li> <li>Direct Membership.</li> <li>A multi-valued list of Subject</li> <li>references, (one of two kinds of</li> <li>Field). The direct members of a</li> </ul>
List	<ul> <li>a Subgroup of a Group, or a</li> <li>member of a Factor Group that</li> <li>contributes positively to a group's</li> <li>membership, has an indirect</li> <li>membership in the group. Also see</li> <li>Direct Membership.</li> <li>A multi-valued list of Subject</li> <li>references, (one of two kinds of</li> <li>Field). The direct members of a</li> <li>group are the values of the group's</li> </ul>
List	<ul> <li>a Subgroup of a Group, or a member of a Factor Group that contributes positively to a group's membership, has an indirect membership in the group. Also see</li> <li>Direct Membership.</li> <li>A multi-valued list of Subject references, (one of two kinds of Field). The direct members of a group are the values of the group's members list. Lists are also used</li> </ul>
List	<ul> <li>a Subject that is a member of a Subject that is a member of a Subgroup of a Group, or a member of a Factor Group that contributes positively to a group's membership, has an indirect membership in the group. Also see Direct Membership.</li> <li>A multi-valued list of Subject references, (one of two kinds of Field). The direct members of a group are the values of the group's members list. Lists are also used to identify which subjects have</li> </ul>
List	<ul> <li>a Subject that is a member of a Subject that is a member of a Subgroup of a Group, or a member of a Factor Group that contributes positively to a group's membership, has an indirect membership in the group. Also see Direct Membership.</li> <li>A multi-valued list of Subject references, (one of two kinds of Field). The direct members of a group are the values of the group's members list. Lists are also used to identify which subjects have which Naming or Access Privileges.</li> </ul>
List	<ul> <li>a Subject that is a member of a <i>Subgroup</i> of a <i>Group</i>, or a member of a <i>Factor Group</i> that contributes positively to a group's membership, has an indirect membership in the group. Also see <i>Direct Membership</i>.</li> <li>A multi-valued list of <i>Subject</i> references, (one of two kinds of <i>Field</i>). The <i>direct members</i> of a group are the values of the group's members list. Lists are also used to identify which subjects have which <i>Naming</i> or <i>Access Privileges</i>.</li> <li>Sites can extend a group type to</li> </ul>
List	<ul> <li>a Subject that is a member of a <i>Subgroup</i> of a <i>Group</i>, or a member of a <i>Factor Group</i> that contributes positively to a group's membership, has an indirect membership in the group. Also see <i>Direct Membership</i>.</li> <li>A multi-valued list of <i>Subject</i> references, (one of two kinds of <i>Field</i>). The <i>direct members</i> of a group are the values of the group's members list. Lists are also used to identify which subjects have which <i>Naming</i> or <i>Access Privileges</i>.</li> <li>Sites can extend a group type to include custom lists; however,</li> </ul>
List	<ul> <li>a Subject that is a member of a <i>Subgroup</i> of a <i>Group</i>, or a member of a <i>Factor Group</i> that contributes positively to a group's membership, has an indirect membership in the group. Also see <i>Direct Membership</i>.</li> <li>A multi-valued list of <i>Subject</i> references, (one of two kinds of <i>Field</i>). The <i>direct members</i> of a group are the values of the group's <b>members</b> list. Lists are also used to identify which subjects have which <i>Naming</i> or <i>Access Privileges</i>.</li> <li>Sites can extend a group type to include custom lists; however, their semantics are external to</li> </ul>
List	<ul> <li>a Subject that is a member of a <i>Subgroup</i> of a <i>Group</i>, or a member of a <i>Factor Group</i> that contributes positively to a group's membership, has an indirect membership in the group. Also see <i>Direct Membership</i>.</li> <li>A multi-valued list of <i>Subject</i> references, (one of two kinds of <i>Field</i>). The <i>direct members</i> of a group are the values of the group's <b>members</b> list. Lists are also used to identify which subjects have which <i>Naming</i> or <i>Access Privileges</i>.</li> <li>Sites can extend a group type to include custom lists; however, their semantics are external to Grouper. See <i>Group</i>.</li> </ul>
List Member	<ul> <li>a Subject that is a member of a <i>Subgroup</i> of a <i>Group</i>, or a member of a <i>Factor Group</i> that contributes positively to a group's membership, has an indirect membership in the group. Also see <i>Direct Membership</i>.</li> <li>A multi-valued list of <i>Subject</i> references, (one of two kinds of <i>Field</i>). The <i>direct members</i> of a group are the values of the group's <b>members</b> list. Lists are also used to identify which subjects have which <i>Naming</i> or <i>Access Privileges</i>.</li> <li>Sites can extend a group type to include custom lists; however, their semantics are external to Grouper. See <i>Group</i>.</li> <li>Any <i>Subject</i> in the membership</li> </ul>
List Member	<ul> <li>a Subject that is a member of a <i>Subgroup</i> of a <i>Group</i>, or a member of a <i>Factor Group</i> that contributes positively to a group's membership, has an indirect membership in the group. Also see <i>Direct Membership</i>.</li> <li>A multi-valued list of <i>Subject</i> references, (one of two kinds of <i>Field</i>). The <i>direct members</i> of a group are the values of the group's <b>members</b> list. Lists are also used to identify which subjects have which <i>Naming</i> or <i>Access Privileges</i>.</li> <li>Sites can extend a group type to include custom lists; however, their semantics are external to Grouper. See <i>Group</i>.</li> <li>Any <i>Subject</i> in the membership list of at least one group. Also,</li> </ul>
List Member	<ul> <li>a Subject that is a member of a <i>Subgroup</i> of a <i>Group</i>, or a member of a <i>Factor Group</i> that contributes positively to a group's membership, has an indirect membership in the group. Also see <i>Direct Membership</i>.</li> <li>A multi-valued list of <i>Subject</i> references, (one of two kinds of <i>Field</i>). The <i>direct members</i> of a group are the values of the group's <b>members</b> list. Lists are also used to identify which subjects have which <i>Naming</i> or <i>Access Privileges</i>.</li> <li>Sites can extend a group type to include custom lists; however, their semantics are external to Grouper. See <i>Group</i>.</li> <li>Any <i>Subject</i> in the membership list of at least one group. Also, a <i>Member</i> of a <i>Group</i> is any</li> </ul>
List Member	<ul> <li>a Subject that is a member of a <i>Subgroup</i> of a <i>Group</i>, or a member of a <i>Factor Group</i> that contributes positively to a group's membership, has an indirect membership in the group. Also see <i>Direct Membership</i>.</li> <li>A multi-valued list of <i>Subject</i> references, (one of two kinds of <i>Field</i>). The <i>direct members</i> of a group are the values of the group's <b>members</b> list. Lists are also used to identify which subjects have which <i>Naming</i> or <i>Access Privileges</i>.</li> <li>Sites can extend a group type to include custom lists; however, their semantics are external to Grouper. See <i>Group</i>.</li> <li>Any <i>Subject</i> in the membership list of at least one group. Also, a <i>Member</i> of a <i>Group</i> is any <i>Subject</i> with a <i>Direct</i> or <i>Indirect</i></li> </ul>
List Member	<ul> <li>a Subject that is a member of a <i>Subgroup</i> of a <i>Group</i>, or a member of a <i>Factor Group</i> that contributes positively to a group's membership, has an indirect membership in the group. Also see <i>Direct Membership</i>.</li> <li>A multi-valued list of <i>Subject</i> references, (one of two kinds of <i>Field</i>). The <i>direct members</i> of a group are the values of the group's <b>members</b> list. Lists are also used to identify which subjects have which <i>Naming</i> or <i>Access Privileges</i>.</li> <li>Sites can extend a group type to include custom lists; however, their semantics are external to Grouper. See <i>Group</i>.</li> <li>Any <i>Subject</i> in the membership list of at least one group. Also, a <i>Member</i> of a <i>Group</i> is any <i>Subject</i> with a <i>Direct</i> or <i>Indirect Membership</i> in the <i>Group</i>.</li> </ul>
List Member	<ul> <li>a Subject that is a member of a <i>Subgroup</i> of a <i>Group</i>, or a member of a <i>Factor Group</i> that contributes positively to a group's membership, has an indirect membership in the group. Also see <i>Direct Membership</i>.</li> <li>A multi-valued list of <i>Subject</i> references, (one of two kinds of <i>Field</i>). The <i>direct members</i> of a group are the values of the group's <b>members</b> list. Lists are also used to identify which subjects have which <i>Naming</i> or <i>Access Privileges</i>.</li> <li>Sites can extend a group type to include custom lists; however, their semantics are external to Grouper. See <i>Group</i>.</li> <li>Any <i>Subject</i> in the membership list of at least one group. Also, a <i>Member</i> of a <i>Group</i>.</li> <li>The direct or <i>Indirect Membership</i>.</li> </ul>
Indirect Membership	<ul> <li>A Subject that is a member of a <i>Subgroup</i> of a <i>Group</i>, or a member of a <i>Factor Group</i> that contributes positively to a group's membership, has an indirect membership in the group. Also see <i>Direct Membership</i>.</li> <li>A multi-valued list of <i>Subject</i> references, (one of two kinds of <i>Field</i>). The <i>direct members</i> of a group are the values of the group's members list. Lists are also used to identify which subjects have which <i>Naming</i> or <i>Access Privileges</i>.</li> <li>Sites can extend a group type to include custom lists; however, their semantics are external to Grouper. See <i>Group</i>.</li> <li>Any <i>Subject</i> in the membership list of a least one group. Also, a <i>Member</i> of a <i>Group</i> is any <i>Subject</i> with a <i>Direct</i> or <i>Indirect Membership</i> in the <i>Group</i>.</li> </ul>
List Member Member	<ul> <li>A Subject that is a member of a <i>Subgroup</i> of a <i>Group</i>, or a member of a <i>Factor Group</i> that contributes positively to a group's membership, has an indirect membership in the group. Also see <i>Direct Membership</i>.</li> <li>A multi-valued list of <i>Subject</i> references, (one of two kinds of <i>Field</i>). The <i>direct members</i> of a group are the values of the group's members list. Lists are also used to identify which subjects have which <i>Naming</i> or <i>Access Privileges</i>.</li> <li>Sites can extend a group type to include custom lists; however, their semantics are external to Grouper. See <i>Group</i>.</li> <li>Any <i>Subject</i> in the membership list of a <i>Group</i> is any <i>Subject</i> with a <i>Direct</i> or <i>Indirect Membership</i>.</li> </ul>
List Member Member	<ul> <li>A Subject that is a member of a subgroup of a Group, or a member of a Factor Group that contributes positively to a group's membership, has an indirect membership in the group. Also see Direct Membership.</li> <li>A multi-valued list of Subject references, (one of two kinds of Field). The direct members of a group are the values of the group's members list. Lists are also used to identify which subjects have which Naming or Access Privileges.</li> <li>Sites can extend a group type to include custom lists; however, their semantics are external to Grouper. See Group.</li> <li>Any Subject in the membership list of a Group is any Subject with a Direct or Indirect Membership in the Group.</li> <li>The direct-only, indirect-only, or direct plus indirect members of a Group. A specific variety of membership is determined by</li> </ul>
List Member Member	<ul> <li>A Subject that is a member of a subgroup of a Group, or a member of a Factor Group that contributes positively to a group's membership, has an indirect membership in the group. Also see Direct Membership.</li> <li>A multi-valued list of Subject references, (one of two kinds of Field). The direct members of a group are the values of the group's members list. Lists are also used to identify which subjects have which Naming or Access Privileges.</li> <li>Sites can extend a group type to include custom lists; however, their semantics are external to Grouper. See Group.</li> <li>Any Subject in the membership list of a Group is any Subject with a Direct or Indirect Membership in the Group.</li> <li>The direct-only, indirect-only, or direct plus indirect members</li> <li>of a Group. A specific variety of membership is determined by context or configuration.</li> </ul>
List Member Member	<ul> <li>A Subject that is a member of a Subject that is a member of a Subject that is a member of a Subject of a Group that contributes positively to a group's membership, has an indirect membership in the group. Also see Direct Membership.</li> <li>A multi-valued list of Subject references, (one of two kinds of Field). The direct members of a group are the values of the group's members list. Lists are also used to identify which subjects have which Naming or Access Privileges.</li> <li>Sites can extend a group type to include custom lists; however, their semantics are external to Grouper. See Group.</li> <li>Any Subject in the membership list of a least one group. Also, a Member of a Group is any Subject with a Direct or Indirect Membership in the Group.</li> <li>The direct-only, indirect-only, or direct plus indirect members of a Group. A specific variety of membership is determined by context or configuration, i.e., the offenult list or a least one.</li> </ul>
List Member Member	<ul> <li>A Subject that is a member of a Group, or a member of a Factor Group that contributes positively to a group's membership, has an indirect membership in the group. Also see Direct Membership.</li> <li>A multi-valued list of Subject references, (one of two kinds of Field). The direct members of a group are the values of the group's members list. Lists are also used to identify which subjects have which Naming or Access Privileges.</li> <li>Sites can extend a group type to include custom lists; however, their semantics are external to Grouper. See Group.</li> <li>Any Subject in the membership list of a Group is any Subject with a Direct or Indirect Membership is any Subject with a Direct or Indirect Membership in the Group.</li> <li>The direct-only, indirect-only, or direct plus indirect members of a Group. A specific variety of membership is determined by context or configuration, i.e., the default User Interface allows the varies and the set of t</li></ul>
List Member Member	<ul> <li>a Subject that is a member of a Subject that is a member of a Subject that is a member of a Subject of a Group, or a membership, has an indirect membership in the group. Also see Direct Membership.</li> <li>A multi-valued list of Subject references, (one of two kinds of Field). The direct members of a group are the values of the group's members list. Lists are also used to identify which subjects have which Naming or Access Privileges.</li> <li>Sites can extend a group type to include custom lists; however, their semantics are external to Grouper. See Group.</li> <li>Any Subject in the membership list of a least one group. Also, a Membership in the Group.</li> <li>The direct-only, indirect-only, or direct plus indirect members of a Group. A specific variety of membership is determined by context or configuration, i.e., the default User Interface allows the user to select among these</li> </ul>
List Member Membership	A Subject that is a member of a Subject that is a member of member of a Factor Group that contributes positively to a group's membership, has an indirect membership in the group. Also see Direct Membership. A multi-valued list of Subject references, (one of two kinds of Field). The direct members of a group are the values of the group's members list. Lists are also used to identify which subjects have which Naming or Access Privileges. Sites can extend a group type to include custom lists; however, their semantics are external to Grouper. See Group. Any Subject in the membership list of at least one group. Also, a Member of a Group is any Subject with a Direct or Indirect Membership in the Group. The direct-only, indirect-only, or direct plus indirect members of a Group. A specific variety of membership is determined by context or configuration, i.e., the default User Interface allows the user to select among these three types of membership where

Naming Privileges	These privileges determine what a <i>Subject</i> can do with a <i>Naming</i> <i>Stem</i> . They are: • CREATE - can create a	Naming privileges are now referred to as Creation privileges and the two types are Create Group (replaces CREATE) and Create Folder (replaces STEM)
	group(s) named with a naming stem, and	
	<ul> <li>STEM - can assign who has CREATE for the naming stem,</li> </ul>	
	and can create naming stems subordinate to this one.	
Naming Stem	A string that forms the leading part of a <b>Group's</b> name. By linking	Stem is a UI "folder"
	the ability to create groups to a	
	<b>CREATE</b> privilege), the possibility	
	that different groups can be given	
	the same name is substantially reduced, and the name of each	
	group can be made to reflect	
	something about the authority	
	see Examples below.	
Stem	A synonym for a <b>Naming Stem</b> .	Stem is a UI "folder"
Subgroup	A <i>Group</i> that is a <i>Direct Member</i> of	F
Subject	An abstraction of any object	Subject is a UI "Entity"
-	whose <i>Memberships</i> are to be	
	Grouper deployments will manage	
	subjects that represent people and	
	groups, but computers, accounts,	
	object maintained in a back-end	
	identity store may be presented as	
	subjects to Grouper by use of the	
Туре	There are two distinct uses for this	
	term in Grouper.	
	Group Type - each Group	
	types associated with it. The	
	Grouper distribution contains	
	support for a single group	
	may register additional types	,
	together with the attributes	
	them, within their Grouper	
	installation. Doing so enables	
	a richer information model	
	or a more diverse set of	
	information models.	
	API v0.2.1 that Grouper 1.0	
	relies on uses the notion	
	of a subject type, such	
	computer", etc.	

# **Examples**

### Step 1: Create a Root Naming Stem

In the example below, a root naming stem is first created. Note: creating a naming stem is required prior to the creation of any groups.

### Naming Stem uofc

attribute	value
naming stem	empty
extension	uofc
displayExtension	The University Of Chicago
name	uofc
displayName	The University Of Chicago

### Step 2: Create a Group

Next, a group may be created using the "uofc" naming stem.

#### Group uofc:exec\_council

attribute	value
naming stem	uofc
extension	exec_council
displayExtension	Executive Council
name	uofc:exec_council
displayName	The University of Chicago: Executive Council

### Step 3: Create a Subordinate Naming Stem and Group

Name and displayName values propagate down through subordinate naming stems, e.g the Biological Sciences Division within U of C:

### Naming Stem uofc:bsd

attribute	value
naming stem	uofc
extension	bsd
displayExtension	Biological Sciences Division
name	uofc:bsd
displayName	The University Of Chicago:Biological Sciences
	Division

Again, a group is created, e.g., the Enterprise Information Systems staff, with the above naming stem, and is displayed as follows:

### Group uofc:bsd:eis\_staff

attribute	value
naming stem	uofc:bsd
extension	eis_staff
displayExtension	Enterprise Information Systems staff
name	uofc:bsd:eis_staff
displayName	The University Of Chicago:Biological Sciences
	Division: Enterprise Information Systems staff

Questions or comments? ① Contact us.

**<u>GROUPER</u>:** <u>About</u> <u>FAQ</u> <u>Software</u> <u>Documentation</u> <u>Archives</u> <u>Contribute</u> <u>WG</u> <u>Contact</u>

# **UI Terminology**

This page last changed on May 16, 2008 by bmk@isc.upenn.edu.

**<u>GROUPER</u>: About FAQ Software Documentation Archives Contribute WG Contact</u>** 

# **Terminology in the Grouper User Interface as of v1.3.0**

The below table breaks terminology into categories and shows the terms used prior to v 1.3, the terms in the production version v1.3.0 and a description of the term

Category	Old Term	New Term	Definition/Description
UI Labels	Privilegees	Enities With Privileges	
	Subject	Entity	An entity is an abstract
			member of a group
			The two most common
			types of entities are
			'person' or 'group'.
			(In the future,
			additional entity
			types may be used to
			describe computers or
	is a direct privilages	has direct privilages	applications.)
	is a indirect privilegee	has indirect privileges	within a group that is a
	is a maneet privilegee	has maneee privileges	member of the group
	Extension	ID	An internal name
			describing this group that
			is generally not exposed
			to the user. This name
			is edited
	Name	ID Path	An internal concatenation
			of the hierarchy to this
			group that is generally not
			exposed to the user
	Display extension	Name	The group name that is
			or searching
	Display name	Path	The path is the
	. ,		concatenation of the
			hierarchy (folders and
			groups) that lead to the
			unique location of this
Hierarchy	stem [conceptual]	Folder	a fundamental unit
	]		(container) of the
			hierarchy that can have
			a parent (folder or 'root')
			or children (folders or
	aroup	aroup	a type of entity made up
	3	3	of members
	Manage Stem	Manage Folder	This is where you can
			create or edit the folders
			within the hierarchy
			hierarchy
Hierarchy Priv	stem [privilege]	Create Folder	the ability to create
			children folders or
			branches in the hierarchy
	Create	Create Group	

			Add or create the name for a new group at this folder (location) in the hierarchy however the entity that creates a group is given Admin rights to the group by default. This does not provide access to manage the group (add membership
	Stem privilege	Creation Privileges	or edit attributes) a hierarchy Is made up of folders. The folder subfolder relationship define the path through
Navigation	saved subjects	Entity Workspace	the hierarchy a session specific area where you can store groups that you will need to create compound
	Saved groups	Group Workspace	groups, etc a session specific area where you can store groups that you will need to create compound groups, etc
	Search subjects	Search	g. cape, etc
Administrative	grouperAll	EveryEntity	Default group privileges that are inherited upon group creation
	GrouperSystem	GrouperSysAdmin	the highest level administrative user of the system
	WheelGroup	SysadminGroup	all people in this group have full system admin
Group Priv	Admin	Admin	Entity (typically group or person) may modify the membership of this group, delete the group or assign
	Member	Member	Any entity (typically group or person) that is a part
	Optin	Optin	Entity (typically group or person) may choose to
	Optout	Optout	Entity (typically group or person) may choose to
	Read	Read	Entity (typically group or person) may see the membership list for this group
	Update	Update	Entity (typically group or person) may modify the membership of this group
	View	View	Entity (typically group or person) may see that this group exists

Below are Grouper concepts described/translated using the UI terminology of version v1.3.0

TERM	DEFINITION
Access Privileges	Privileges that determine what a <b>Entity</b> can do with a
	Group. They are:
Attribute	<ul> <li>ADMIN - can assign access privileges and manage all group information,</li> <li>UPDATE - can manage membership of the group (implies READ),</li> <li>READ - can see the membership of the group (implies VIEW), and</li> <li>VIEW - can see the group. In addition, a group may have options for its members to:</li> <li>OPTIN - can add self to the membership, and</li> <li>OPTOUT - can remove self from membership.</li> <li>A single-valued string associated with a <i>Group</i> or a <i>Folder</i>. By default, Grouper supports six attributes (one of two kinds of <i>Field</i>):</li> </ul>
	• <b>UUID</b> - a Grouper-assigned, globally unique identifier.
Composite Group	<ul> <li>ID- the relative name of the group or folder within its parent folder; the contribution of a single element, such as a group or a folder, to the cumulative name.</li> <li>ID Path- used to facilitate searching for groups by name, it is a read-only string representation of the logical ordered pair of (<i>parent folder, ID</i>). This attribute is systemmaintained. The string representation of the ID Path attribute is: &lt;<i>folder&gt;:<id></id></i>.</li> <li>Name- a displayed form of the ID.</li> <li>Path -used to facilitate searching for groups by the path, it is a read-only string representation of the logical ordered pair of <i>folder, Name</i>. This attribute is systemmaintained. The string representation of the ID.</li> <li>Path -used to facilitate searching for groups by the path, it is a read-only string representation of the logical ordered pair of <i>(Path of parent folder, Name)</i>. This attribute is systemmaintained. The string representation of the path attribute is: <i><path folder="" of="" parent="">: <name></name></path></i>.</li> <li>description - a description of the group or folder.</li> </ul>
	<ul> <li>union - all entities must be a member of one OR the other factor group, e.g., Group Z = members of either Group X OR Group Y, or Z = X U Y.</li> <li>intersection - all entities that are members of the first factor group AND the second factor group, e.g. Group Z = members of both Group X AND</li> </ul>
	<ul> <li>Group Z = members of both Group X AND Group Y, or Z = X # Y.</li> <li>relative complement - all members of the first factor group that are NOT members of the second factor group.</li> <li>e.g., Group Z = members of Group X AND NOT Group Y, or Z = X - Y.</li> </ul>

Direct Membership	An entity that is listed in the <b>Membership</b> list of a <b>Group</b> has a direct membership in the group. Also see <b>Indirect Membership</b> .
Factor Group	A Group in combination (union, intersection, or relative complement) with that of another factor group, which defines the membership of a resulting Composite Group.
Field	Either an <b>Attribute</b> or a <b>List</b> . Grouper groups are a collection of attributes and lists, i.e., a collection of fields. The set of fields attached to a given group is a function of the set of <b>Group Types</b> it has been assigned.
Group	A list of <b>Subjects</b> having <b>Membership</b> in the group, together with other attributes about the group. A list can have zero or more entries. In Grouper, a list contains only entity references, and an attribute is a single-valued string. A group must be created in an existing <b>Folder</b> . If a group is made a member, i.e., a <b>Subgroup</b> , of another group, the members of the group will also be made members. By default, a Grouper group has:
	<ul> <li>six naming <i>Attributes</i>,</li> <li>a description attribute, and</li> <li>a members list. This information model can be extended to include additional site-defined attributes and lists.</li> </ul>
Group Math	Any combination of groups for the purpose of creating another group based on the memberships of those groups. See <b>Composite Group</b> .
Indirect Membership	An <i>Entity</i> that is a member of a <i>Subgroup</i> of a <i>Group</i> , or a member of a <i>Factor Group</i> that contributes positively to a group's membership, has an indirect membership in the group. Also see <i>Direct</i> <i>Membership</i> .
List	A multi-valued list of <b>Entity</b> references, (one of two kinds of <b>Field</b> ). The <b>direct members</b> of a group are the values of the group's <b>members</b> list. Lists are also used to identify which entities have which <b>Creation</b> or <b>Access Privileges</b> . Sites can extend a group type to include custom lists; however, their semantics are external to Grouper. See <b>Group</b> .
Member	Any <b>Entity</b> in the membership list of at least one group. Also, a <b>Member</b> of a <b>Group</b> is any <b>Entity</b> with a <b>Direct</b> or <b>Indirect Membership</b> in the <b>Group</b> .
Membership	The direct-only, indirect-only, or direct plus indirect members of a <b>Group</b> . A specific variety of membership is determined by context or configuration, i.e., the default User Interface allows the user to select among these three types of membership where appropriate.
Creation Privileges	<ul> <li>These privileges determine what an <i>Entity</i> can do with a <i>Folder</i>. They are:</li> <li>CREATE GROUP - can create a group(s) named with a naming stem, and</li> <li>CREATE FOLDER - can assign who can CREATE folders (and sub folders) in a branch of</li> </ul>
Path	A string that precedes the <b>Group's</b> name. By linking the ability to create groups to a specified folder (via the <b>Creation</b> privilege), the possibility that different groups can be given the same name is substantially reduced, and the name of each group can be made

	to reflect something about the authority under which it was created. see Examples below.
Subgroup Entity	A <b>Group</b> that is a <b>Direct Member</b> of another group. An abstraction of any object whose <b>Memberships</b> are to be managed by Grouper. Most Grouper deployments will manage entities that represent people and groups, but computers, accounts, services, or any other type of object maintained in a back-end identity store may be presented as an entity to Grouper by use of the Subject API.
Туре	<ul> <li>There are two distinct uses for this term in Grouper.</li> <li>Group Type - each Group has one or more group types associated with it. The Grouper distribution contains support for a single group type called "base", but sites may register additional types, together with the attributes and lists associated with them, within their Grouper installation. Doing so enables management of groups with a richer information model or a more diverse set of information models.</li> <li>Entity Type - the Subject API v0.2.1 that Grouper 1.3 relies on uses the notion of a subject type, such as "person", "group", or "computer", etc.</li> </ul>

# Examples

### Step 1: Create a Root Folder

In the example below, a root Folder is first created. Note: creating a folder is required prior to the creation of any groups.

### Folder <u>uofc</u>

attribute	value
folder	empty
ID	uofc
name	The University Of Chicago
ID path	uofc
path	The University Of Chicago

### Step 2: Create a Group

Next, a group may be created using the "uofc" naming stem.

### Group uofc:exec\_council

attribute	value
folder	uofc
ID	exec_council
name	Executive Council
ID path	uofc:exec_council
path	The University of Chicago: Executive Council

### Step 3: Create a Subordinate Folder and Group

Folder ID and Path values propagate down through subordinate floders, e.g the Biological Sciences Division within U of C:

### Folder uofc:bsd

attribute	value
folder	uofc
ID	bsd
name	Biological Sciences Division
ID path	uofc:bsd
path	The University Of Chicago:Biological Sciences
	Division

Again, a group is created, e.g., the Enterprise Information Systems staff, with the above folder, and is displayed as follows:

### Group uofc:bsd:eis\_staff

attribute	value
folder	uofc:bsd
ID	eis_staff
name	Enterprise Information Systems staff
ID path	uofc:bsd:eis_staff
path	The University Of Chicago:Biological Sciences
	Division: Enterprise Information Systems staff

Questions or comments? <sup>1</sup> Contact us.

GROUPER: About FAQ Software Documentation Archives Contribute WG Contact

### **Grouper Binary Release**

This page last changed on Nov 22, 2008 by mchyzer.

The grouper binary release is a package which does not need to be built with ant.

### Contents

- Grouper jar
- All jars that grouper requires
- Configuration files (from examples, need to be customized)
- Example jdbc jars (for Oracle, MySQL, Postgres, hsql)
- Grouper javadoc
- Grouper shell (GSH) script used to interact with grouper from the command line

#### Using

- You can use the binary release of grouper to get started with grouper, or to upgrade a current installation. Note, if you want to run the UI, you need to build it, and grouper itself
- Unzip the binary release
- Run from command prompt: bin/gsh -registry -runscript
- This will create the database in hsql
- Run from command prompt: bin/gsh
   This will run Grouper Shell
- Type exit, edit the conf/grouper.hibernate.properties with the hibernate dialect, driver, url, user, and pass of a real database (preferably oracle, mysql, or postgres)
- Run from command prompt: bin/gsh -registry -runscript
- This will create the database in hsql
- Run from command prompt: bin/gsh
   This will run Crowner Shell
  - This will run <u>Grouper Shell</u>
- Go through the conf/grouper.properties, conf/sources/xml, etc and look at options available and customize

### **Grouper Design Guidelines**

This page last changed on Jul 18, 2007 by jbibbee@internet2.edu.

GROUPER: About FAQ Software Documentation Contribute WG Contact

This document will offer several considerations as you prepare to deploy Grouper at your campus.

... more to come soon!

Questions or comments? <sup>1</sup> Contact us.

GROUPER: About FAQ Software Documentation Contribute WG Contact

### **Grouper - Loader**

This page last changed on Jan 08, 2009 by mchyzer.

#### Grouper loader v1.4.0

Here is a grouper loader which can be used to automatically manage grouper memberships based on a data source.

Penn is using it in production to load membership for groups, and for groups of groups (in Penn's case, org lists).

#### One-time setup in your grouper database

To make a dynamic (loadable) group, first you need the correct metadata in grouper. The easiest way is to set the grouper-loader.properties key loader.autoadd.typesAttributes to true. If you dont want to do that, then here is the setup in GSH:

```
subj=SubjectFinder.findById("GrouperSystem")
sess=GrouperSession.start(subj)
type=GroupType.createType(sess, "grouperLoader")
read=Privilege.getInstance("read")
admin=Privilege.getInstance("admin")
type.addAttribute(sess, "grouperLoaderType", read, admin, true)
type.addAttribute(sess, "grouperLoaderDbName", read, admin, true)
type.addAttribute(sess, "grouperLoaderScheduleType", read, admin, true)
type.addAttribute(sess, "grouperLoaderQuery", read, admin, true)
type.addAttribute(sess, "grouperLoaderQuartzCron", read, admin, false)
type.addAttribute(sess, "grouperLoaderIntervalSeconds", read, admin, false)
type.addAttribute(sess, "grouperLoaderPriority", read, admin, false)
type.addAttribute(sess, "grouperLoaderAndGroups", read, admin, false)
type.addAttribute(sess, "grouperLoaderGroupTypes", read, admin, false)
type.addAttribute(sess, "grouperLoaderGroupsLike", read, admin, false)
type.addAttribute(sess, "grouperLoaderGroupQuery", read, admin, false)
```

Note that a loadable group has the type "grouperLoader", and there are some attributes that you can set about the group:

- grouperLoaderType: there will be various types to choose from, currently only SQL\_SIMPLE is available, which is a group whose membership is fed from a query, and the whole query's results will be the groups results (not incremental). SQL\_GROUP\_LIST is available and requires a group\_name column in query, so one query can control multiple group memberships. This will default to SQL\_SIMPLE if no group name before the FROM in the query. Else it will be SOL\_GROUP\_LIST.
- grouperLoaderDbName: if it is a sql based type, this is the name in the grouper-loader.properties of the db connection properties. If this is set to "grouper" that is a special reserved term for the grouper db (in grouper.hibernate.properties) Here is a snippet from grouper-loader.properties

# specify the db connection with user, pass, url, and driver class # the string after "db." is the name of the connection, and it should not have # spaces or other special chars in it db.warehouse.user = mylogin db.warehouse.pass = secret db.warehouse.url = jdbc:mysql://localhost:3306/grouper db.warehouse.driver = com.mysql.jdbc.Driver

- grouperLoaderScheduleType: Grouper-loader uses the quartz open source job scheduler, and currently supports two schedule types (note, that a job will not start if a previous run has not finished. This defaults to CRON if there is a CRON filled in, else it defaults to START\_TO\_START\_INTERVAL
  - ° CRON: This is a cron-like syntax that I think is quartz specific
  - START\_TO\_START\_INTERVAL: This is a repeated schedule that runs based on a delay from the start of one run to the start of another run
- grouperLoaderQuery: This is the query to run in the DB, which must have certain columns required or optional based on the grouperLoaderType. e.g. for SQL\_SIMPLE, the SUBJECT\_ID is required, and the SUBJECT\_SOURCE\_ID is optional. If your DB supports views, might not be a bad idea to link query up to a view so you can easily see what it will return and change it without affecting the group attribute. But will work with any select query. This is required

- grouperLoaderQuartzCron: If a CRON schedule type, this is the cron setting string from the quartz product to run a job daily, hourly, weekly, etc: <u>http://www.opensymphony.com/quartz/wikidocs/</u> <u>TutorialLesson6.html</u>
- grouperLoaderIntervalSeconds: If a START\_TO\_START\_INTERVAL schedule type, this is the number of seconds between the start of one run to the start of another run. This defaults to daily if not filled in. Note, for daily jobs, it is probably better to use cron so it wont fire up each time the loader is restarted.
- grouperLoaderPriority: Quartz has a fixed threadpool (max configured in the grouperloader.properties), and when the max is reached, then jobs are prioritized by this integer. The higher the better, and the default if not set is 5.
- grouperLoaderAndGroups: If you want to restrict membership in the dynamic group based on other group(s), put the list of group names here comma-separated
- grouperLoaderGroupTypes: whatever you put in the value should be comma separated GroupTypes which will be applied to the loaded groups. The reason this enhancement exists is so we can do a SQL\_GROUP\_LIST query and attach addIncludeExclude to the groups. Note, if you do this (or use some requireGroups), the group name in the loader query should end in the system of record suffix, which by default is \_systemOfRecord.
- grouperLoaderGroupsLike attribute: this should be a sql like string (e.g. school:orgs:%org %\_systemOfRecord), and the loader should be able to query group names to see which names are managed by this loader job. So if a group falls off the loader resultset (or is moved), this will help the loader remove the members from this group. Note, if the group is used anywhere as a member or composite member, it wont be removed. All include/exclude/requireGroups will be removed. Though the two groups, include and exclude, will not be removed if they have members. There is a grouper-loader.properties setting to note remove loader groups if empty and not used:

#if using a sql table, and specifying the name like string, then shoudl the group (in addition to memberships)

# be removed if not used anywhere else?
loader.sqlTable.likeString.removeGroupIfNotUsed = true

 grouperLoaderGroupQuery: query (optional) for SQL\_GROUP\_LIST which should return cols: group\_name, group\_display\_name (optional), group\_description (optional) which if there are used for the group display and and extension. Note: the parent stem display names are only changed when creating them. This should return all groups in the membership list, and if not there, its ok, the extension will be used as display extension, and a generated description. Note: the display name is the display path, with the display extension of each parent stem.

# Configure a loadable group (obviously any number of dynamic loadable groups can exist at once)

With GSH, it would look like this:

```
group=getGroups("aStem:aGroup2")
groupAddType("aStem:aGroup2", "grouperLoader")
setGroupAttr("aStem:aGroup2", "grouperLoaderDbName", "grouper")
setGroupAttr("aStem:aGroup2", "grouperLoaderType", "SQL_SIMPLE")
setGroupAttr("aStem:aGroup2", "grouperLoaderScheduleType", "START_TO_START_INTERVAL")
setGroupAttr("aStem:aGroup2", "grouperLoaderQuery", "select SUBJECT_ID, SUBJECT_SOURCE_ID from
agroup2_v")
setGroupAttr("aStem:aGroup2", "grouperLoaderIntervalSeconds", "30")
```

You can also use the UI, here are screenshots (obviously these need some work).

	🧶 cugʻin 🖓 dau su 🖉 daaris 🕰 examin	🛐 USIIY I 🕴 🔁 🐨	🖓 i ago 👷 i con 🐨
INTERNET			Groupe
	Welco	NICHAEL CHRISTOPHER HYZER 🛛 🤗 Log	out Artias self 🛛 💌 Chang
My enrollment	EXPLORE		
My memberships	Edit group 🛽		
Join groups			
My responsibilities Manage groups	Current location is: PRoot: a stem: MaGroup2		
create groups	Name	aGroup2	
My tools Explore	ID	aGroup2	
Search	Description	some description	
Group workspace Entity workspace	Assign privileges to everyone	🗆 admin 🗆 update 🗹 read 🗹 view [	🗌 optin 🔲 optout
Help	Select group types	🗌 dynamic 🗹 grouperLoader	
	Save Back to group summary		
Grouper is sponsored by			

SITY OF PENNSYLVANIA		Ŗ	Gr
	Welcome Hyzer, Chris : mchyzer, Staff, ISC	Administrative Systems Tools and Techno	ologies, PROGRAMMER ANALYST SR 🛛 🛛 Log out 🛛 Act as admin 🦄
	EXPLORE		
	Edit attributes		
space	Current location is: Proot: Pa stem: MaGroup2		
space	Group type	Attribute	Yalue
	grouperLoader		
		grouperLoaderDbName	grouper
		grouperLoaderIntervalSeconds	30
		grouperLoaderPriority	
		grouperLoaderQuartzCron	
		grouperLoaderQuery	select SUBJECT_ID from agroup2_v
		grouperLoaderScheduleType	START_T0_START_INTERVAL
		grouperLoaderType	SQL_SIMPLE
	Save attributes and finish Save attribu	ites and add members	
oonsored by			

### Run grouper loader

The first time you run, it will probably fail, and give you DDL in the logs to run in your database (to add a couple of tables). Run the scripts and you should be all set.

Run with:

GROUPER\_HOME/bin/gsh.sh -loader

This will kick off as a command line program that you will want to run as a service. This process will be always running, the scheduler will schedule the jobs. You should monitor the process with a monitoring tool like nagios or whatever you use at your institution so that you know when it is not up.

You can also run a one-timer via gsh. This is useful to run once at the beginning, and not have to wait for the schedule. Or to troubleshoot e.g.

loaderGroup = GroupFinder.findByName(GrouperSession.startRootSession(), "school:orgs:orgGroup");

### Logging of jobs in DB

Each job (and subjob if the job manages multiple things) will have an entry in the grouploader\_log table. This will show the following information. This can be used to tune performance problems, see which jobs have unresolvable subjects, verify that jobs are running, etc.

COLUMN NAME DATA\_TYPE ID VARCHAR2 JOB\_NAME VARCHAR2 STATUS VARCHAR2 STARTED\_TIME TIMESTAMP(6) ENDED\_TIME TIMESTAMP(6) MILLIS NUMBER NUMBER MILLIS GET DATA MILLIS\_LOAD\_DATA NUMBER JOB TYPE VARCHAR2 JOB\_SCHEDULE\_TYPE VARCHAR2 JOB\_DESCRIPTION VARCHAR2 JOB MESSAGE VARCHAR2 HOST VARCHAR2 GROUP\_UUID VARCHAR2 VARCHAR2 JOB\_SCHEDULE\_QUARTZ\_CRON JOB\_SCHEDULE\_INTERVAL\_SECONDS NUMBER NUMBER JOB\_SCHEDULE\_PRIORITY LAST\_UPDATED TIMESTAMP(6) NUMBER UNRESOLVABLE\_SUBJECT\_COUNT INSERT\_COUNT NUMBER UPDATE\_COUNT NUMBER DELETE\_COUNT NUMBER TOTAL\_COUNT NUMBER PARENT\_JOB\_NAME VARCHAR2 PARENT\_JOB\_ID VARCHAR2

You can also look at log4j debug log messages, and info log messages (less frequent). to see these, set log level in log4j.properties

## Log debug info on loader to see progress etc log4j.logger.edu.internet2.middleware.grouper.app.loader = INFO -orlog4j.logger.edu.internet2.middleware.grouper.app.loader = DEBUG

#### Misc

- You can set transaction level in the grouper-loader.properties. It defaults to not use transaction since for huge groups, you might have memory or db problems
- By default only wheel group members can edit the grouperLoader type or attributes. You can edit these settings in the grouper.properties in the type security part.

#### Possible to do's

- add subject souce to group attribute (or default at least) so it doesnt have to be a sql column if all are the same
- make specific blackout times (runtime via config file?)
- make jobs based on person trigger. If there is a query that says when people change, then update that person's memberships in the groups that are dynamic based on that person-change-query
- make full refresh jobs for the incremental jobs (e.g. weekly)
- save quartz info to DB so that stopping / starting isnt that drastic
- make more job types (jndi?)

- make an RMI server so that interactions can happen at runtime (to see status, stop/start jobs, etc). Maybe this would happen from gsh
- try the name pattern after loader is done, and if the number of groups is less than the number of groups in this round of loader, set the job status to WARNING and add a descriptive message

### Comments

Below I'll contrast this approach with the one I've been using - and modifications I would expect to make going forward. This is not to say my approach is better or would be more generally applicable:

#### **Custom attributes**

Custom attributes are expensive. I have used a few for loading specifically and lots more for course codes etc. Going forward, where practical, it might make sense to either use one or two attributes which encode multiple values - the fewer rows returned the better.

Alternatively, having a single marker attribute and using a separate table to hold additional configuration may make sense. This would assume you wouldn't generally search on this additional configuration, but a loader would be able to retrieve it. Looking at the example it seems likely that many groups would share this information i.e. loader maintained groups might fall into a 'small' number of categories i.e. student records, personnel ..., so you could load the config info once for each category.

#### Number of queries

For each category of groups I have a single query which returns subject\_id and group\_name for all groups. I build a nested Map of these results. I also have one query which retrieves all (non-group) immediate memberships for all groups. I also turn this into a nested Map and compute the differences between current and desired memberships so I only add / remove members as needed.

Going forward, rather than query all immediate memberships I would aim to get all memberships for a given category.

Currently I query the Grouper database directly, however, it would be possible to extend the API to do these types of queries.

I don't use transactions currently but would expect to keep all changes to a single group in a transaction.

### Creating groups and assigning privileges

My loader comes in two parts - the membership loader described above and one which creates/updates stems/groups along with loader maintained types / attributes and also lists of privileged subjects - generally groups themselves. I use some custom attributes and a custom UIGroupPrivilegeResolver implementation so that the UI only allows users (including GrouperSystem) to maintain allowed attributes i.e. ones which would be overwritten with the next scheduled load.

Due to the lack of transactions I created Pseudo stems/groups and members so I could attempt to assemble parallel objects - detecting any errors before invoking API 'write' methods. I drive the loading with parameterised queries which will provide a row for each group of a category, or an individual group or a range of groups. For each row I build the pseudo objects (custom coded for each category type) and 'write' them.

I chose this approach because if there is an error part way through it is possible to start again where the error occurred, or even to run an individual group and debug a problem. Previously I had used a recursive approach which would start building faculties, departments and courses. It was difficult to pick up from where you left off if you got an out of memory error.

Going forward I don't think it will be necessary to use pseudo objects as I should be able to wrap write operations with transactions. In addition I would like to come up with a more generic means of defining what work needs to be done for each category of groups i.e. for each course I make admin / staff / student groups and set up privileges and attributes. I would like to have an XML configuration which acts as a template and drives the process.

### Scheduling

I don't at the moment - and it is some time since I did a load, so using Quartz might help.

Posted by gary.brown@bristol.ac.uk at Apr 28, 2008.

## GrouperShell (gsh)

This page last changed on Jan 25, 2009 by tbarton@uchicago.edu.

GROUPER: About FAQ Software Documentation Archives Contribute WG Contact

# GrouperShell (gsh) as of v1.4.0

gsh is a command line shell for administering and interacting with the Grouper API. It can be used in both a batch and interactive manner. It is built on <u>Java BeanShell</u>

### **API Compability**

gsh is now a core part of the Grouper API and so is always compatible with the current release.

### Installation

When using the Grouper API source distribution, grouper.jar needs to be built before using gsh.sh for the first time:

cd \$GROUPER\_HOME ant dist

### Usage

### For Windows use \$GROUPER\_HOME\bin\gsh.bat

Run gsh as an interactive shell:

\$GROUPER\_HOME/bin/gsh.sh

#### Read gsh commands from STDIN:

\$GROUPER\_HOME/bin/gsh.sh -

#### Read gsh commands from a script file:

\$GROUPER\_HOME/bin/gsh.sh /path/to/your/script.gsh

#### Run Grouper utilities:

```
$GROUPER_HOME/bin/gsh.sh <option>
args: -h,
                       Prints this message
args: -check,
                       Performs startup check and enters an
                        interactive shell
args: -runarg <command> Run command (use \\n to separate commands)
args: -main <class> [args...]
  class,
                       Full class name (must have main method)
  args,
                       args as required by main method of class
args: -initEnv [<configDir>]
      On Windows sets GROUPER_HOME and adds GROUPER_HOME/bin to path
       For *nix 'source gsh.sh' for the same result
       configDir optionally adds an alternative conf directory than
       GROUPER_HOME/conf to the classpath
args: (-xmlimport | -xmlexport | -loader | -test | -registry | -usdu |
      findbadmemberships)
                        Enter option to get additional usage for that
                        option
  -xmlimport,
                       Invokes XmlExporter
  -xmlexport,
                       Invokes XmlImporter
  -loader,
                        Invokes GrouperLoader
  -registry,
                        Manipulate the Grouper schema and install
                       bootstrap data
                        Run JUnit tests
  -test,
                        Invoke USDU - Unresolvable Subject Deletion
  -usdu,
                        Utility
```

-findbadmemberships, Check for membership data inconsistencies

### Supported Commands

#### **Grouper API methods**

Any Grouper API method can be directly invoked just by referencing it, inclusive of the class in which it is defined. Methods return a java object which can be stored in a variable. For example, the following gsh session determines all of the groups to which a given subject belongs:

```
gsh 0% subj = findSubject("SD00125")
subject: id='SD00125' type='person' source='kitn-person' name='Barton, Tom'
gsh 1% sess = GrouperSession.start(subj)
edu.internet2.middleware.grouper.GrouperSession: 29c40f97-9fb0-4e45-88bc-
a14877a6c9b5,'SD00125','person'
gsh 2% member = MemberFinder.findBySubject(sess, subj)
member: id='SD00125' type='person' source='kitn-person' uuid='d0fa765e-1439-4701-89b1-9b08b4ce9daa'
gsh 3% member.getGroups()
group: name='etc:sysadmingroup' displayName='Grouper Administration:SysAdmin Group' uuid='6f77fb36-
b466-481a-84a7-7af609f1ad09'
```

#### Groups

Command	Description
addGroup(parent stem name, extension,	Add group to registry
displayExtension)	
delGroup(name)	Delete group from registry
getGroupAttr(group name, attr)	Get value of group attribute
getGroups(name)	Find all groups with a matching naming attribute
	value, returns a Set of groups
setGroupAttr(group name, attr, value)	Set value of group attribute
GroupFinder.findByName(grouperSession, name)	Find one group by name
GroupFinder.findByUuid(grouperSession, name)	Find one group by uuid

### **Group Types**

Command	Description
groupAddType(group name, type name)	Add type to group
groupDelType(group name, type name)	Delete type from group
groupGetTypes(group name)	Get group's types
groupHasType(group name, type name)	Check whether group had type
typeAdd(type name)	Create custom group type
typeAddAttr(type name, attr name, read, write,	Create custom group attribute. read and
required)	<i>write</i> must be an AccessPrivilege (e.g.
	AccessPrivilege.ADMIN)
typeAddList(type name, attr name, read, write)	Create a custom list. read and write must be an
	AccessPrivilege (e.g. AccessPrivilege.ADMIN).
typeDel(type name)	Delete group type
typeDelField(type name, field name)	Delete custom field from group type
typeFind(type name)	Find the group
typeGetFields(type name)	Get fields associated with the group type

#### Member change subject

Change subject of a Member object, e.g.:

grouperSession = GrouperSession.startRootSession(); oldSubject = findSubject("10021368"); member = MemberFinder.findBySubject(grouperSession, oldSubject); newSubject = findSubject("10021366"); member.changeSubject(newSubject);

member.changeSubject(newSubject);	Change the subject of the member object. If the
	subject is the same, its a no-op. If the new subject
	does not have a Member object, then the existing
	member object simply gets new subject information.
	If the new subject does have a member object,
	then all objects in the grouper registry which uses
	the old member, will be updated to the new member.
	Then the old member object is deleted from the
	registry
member.changeSubject(newSubject,!	Change the subject, but dont delete the old member.
Member.DELETE_OLD_MEMBER);	Do this if the way which deletes the old member
	doesnt work due to foreign keys. This will do all the
	work it can, and the rest can be manual
member.changeSubjectReport(newSubject,Member.l	DENDETED COUNTLY MEMBERROF, just print a report to the
	screen of what will be done. Dry-run.

### Memberships

Command	Description
addComposite(group name, composite type, left	Add composite membership. e.g.
group name, right group name)	CompositeType.UNION
addMember(group name, subject id)	Add member to the members list for the group.
addMember(group name, subject id, field)	Add member to the specified list for the group.
delComposite(group name)	Delete composite membership from group
delMember(group name, subject id)	Delete member from the members list for the group
delMember(group name, subject id, field)	Delete member from the specified list for the group
getMembers(group name)	Get members of group
hasMember(group name, subject id)	Check whether subject is member of the members list
hasMember(group name, subject id, field)	Check whether subject is member of the specified list

### Privileges

Command	Description
grantPriv(group name, subject id, privilege)	Grant privilege on group. privilege must be an
	AccessPrivilege (e.g. AccessPrivilege.ADMIN)
grantPriv(stem name, subject id, privilege)	Grant privilege on stem. privilege must be a
	NamingPrivilege (e.g. NamingPrivilege.STEM)
hasPriv(group name, subject id, privilege)	Check whether subject has privilege on group.
	<i>privilege</i> must be an <i>AccessPrivilege</i> (e.g.
	AccessPrivilege.ADMIN)
hasPriv(stem name, subject id, privilege)	Check whether subject has privilege on strem.
	<i>privilege</i> must be a <i>NamingPrivilege</i> (e.g.
	NamingPrivilege.STEM)
revokePriv(group name, subject id, privilege)	Revoke privilege on group. <i>privilege</i> must be an
	AccessPrivilege (e.g. AccessPrivilege.ADMIN)
revokePriv(stem name, subject id, privilege)	Revoke privilege on stem. privilege must be a
	NamingPrivilege (e.g. NamingPrivilege.STEM)

### Registry

Command	Description
registryInitializeSchema()	Will generate schema DDL for the DB, and wont drop
	before creating, will not run script
registryInitializeSchema(registryInitializeSchema.DR	@enerters
	will not run script
registryInitializeSchema.WRITE_AND_RUN_SCRIPT)	generate DDL for the DB, not dropping, but will run the script after writing it to file

registryInitializeSchema(registryInitializeSchema.DR <b>@<u>en</u>ērHHs_DIREA</b> diEthe DB, drop existing grouper	
	tables, and run the script after writing it to file
registryInitializeSchema.WRITE_AND_RUN_SCRIPT)	
resetRegistry()	Restore registry to default state(delete data from all
	tables, install defaults)
registryInstall()	If the default Grouper data is not there, it will be
	added (e.g. root stem, default fields, etc)

### Stems

Command	Description
addRootStem(extension, displayExtension)	Add top-level stem to the registry
addStem(parent stem name, extension, displayExtension)	Add stem to registry
delStem(stem name)	Delete stem from registry
getStemAttr(stem name, attr)	Get value of stem attribute
getStems(name)	Find all stems with a matching naming attribute value, returns a Set of stems
setStemAttr(stem name, attr, value)	Set value of stem attribute
StemFinder.findByName(grouperSession, name) StemFinder.findByUuid(grouperSession, uuid)	Find one stem by name Find one stem by uuid

### Subjects

Command	Description
addSubject(id, type, name)	Add local subject to registry
findSubject(id)	Find a subject
findSubject(id, type)	Find a subject
findSubject(id, type, source)	Find a subject
getSources()	Find all Subject sources

### System

Command	Description
sqlRun(file)	Execute each line of a sql file, just like ant
	would. This can run the files generated by
	registryInitializeSchema()
sqlRun(string)	Executes a single sql statement
exit	Terminate shell
help()	Display usage information
history()	Print commands that have been run
history(N)	Print the last N commands that have been run
last()	Run the last command executed
last(N)	Execute command number N
p(command)	Pretty print results. This command is more useful
	when GSH_DEVEL is enabled
quit	Terminate shell
version()	Return version information

### Unresolvable subject deletion utility (USDU)

usdu finds which memberships are with subjects which cannot be found in a subject source, and prints them on the screen

- if the usdu.DELETE option is passed in, then the memberships will be deleted

- a grouper session must be open when this command is run.

For more information, see Unresolvable Subject Deletion Utility (USDU)

Command	Description
<pre>subject=SubjectFinder.findById("GrouperSystem") session=GrouperSession.start(subject) usdu()</pre>	Sample call to find all unresolvable subjects in the registry and print details to the screen

usdu(usdu.DELETE)	Pass in that you want to delete memberships in the usdu call
usduBySource("schoolperson")	Work only in a specific subject source, pass in the sourceId from sources.xml
usduBySource("schoolperson", usdu.DELETE)	Work in a specific source and delete membeships
<pre>subject=SubjectFinder.findById("GrouperSystem")</pre>	Work only with a specific member
session=GrouperSession.start(subject)"	
memberSubject=SubjectFinder.findById("1234567"	)
member=MemberFinder.findBySubject(session,mem	hberSubject)
usduByMember(member)	
usduByMember(member, usdu.DELETE)	usdu by member, and delete memberships

### Find bad memberships

This command will find membership records in the database which are invalid, and prints them on the screen, along with a GSH script that will fix the memberships.

For more information, see **Bad Membership Finder Utility** 

Command	Description
findBadMemberships()	complete findBadMemberships run
subject=SubjectFinder.findById("GrouperSystem")	find bad naming privileges for a specific stem
session=GrouperSession.start(subject)	
<pre>stem = StemFinder.findByName(session, "test")</pre>	
findBadMembershipsByStem(stem)	
subject=SubjectFinder.findById("GrouperSystem")	find bad memberships and access privileges for a
session=GrouperSession.start(subject)	specific group
group = GroupFinder.findByName(session,	
"test:testGroup")	
findBadMembershipsByGroup(group)	

### XML legacy

Command	Description
xmlFromFile(filename)	Load registry from XML in file
xmlFromString(xml)	Load registry from XML in string
xmlFromURL(url)	Load registry from XML at URL
xmlToFile(filename)	Exports registry to file
xmlToString()	Exports registry to string.
xmlUpdateFromFile(filename)	Update registry from XML in file
xmlUpdateFromString(xml)	Update registry from XML in string
xmlUpdateFromURL(url)	Update registry from XML at URL

### XML export

There is an object: XmlExport which has various chaining methods, which should be ended with an exportTo() method. You can export to file or string.

For more information, see Import-Export

Command	Description
XmlExport xmlExport.stem(stem)	The stem to export. Defaults to the ROOT stem.
XmlExport xmlExport.group(group)	The group to export
XmlExport xmlExport.relative(boolean)	If group or stem specified do not export parent Stems.
XmlExport xmlExport.includeParent(boolean)	If group specified, export from the parent stem
XmlExport xmlExport.childrenOnly(boolean)	If stem specified, export child stems and groups only - not the specified stem
XmlExport xmlExport.userProperties(file)	Properties file for extra settings for import
XmlExport	Operate within a certain grouper session (defaults to
xmlExport.grouperSession(grouperSession)	root session)
void xmlExport.exportToFile(file)	Export to an XML file

#### void xmlExport.exportToString(string)

#### Export to an XML string

#### Examples:

gsh 1% new XmlExport().exportToFile(new File("c:/temp/export.xml"))
gsh 1% grouperSession = GrouperSession.start(SubjectFinder.findById("mchyzer"));
gsh 2% stem = StemFinder.findByName(grouperSession, "aStem");
gsh 3% new XmlExport().stem(stem).relative(true).userProperties(new File("C:/temp/

some.props")).grouperSession(grouperSession).exportToFile(new File("c:/temp/export.xml"));

#### -or- (without chaining)

```
gsh 3% xmlExport = new XmlExport();
gsh 4% xmlExport.stem(stem);
gsh 5% xmlExport.grouperSession(grouperSession);
gsh 6% xmlExport.exportToFile(new File("c:/temp/export.xml"))
```

#### XML import

There is an object: XmlImport which has various chaining methods, which should be ended with an importFrom() method. You can import from file, string, or url.

For more information, see Import-Export

Command	Description
XmlImport xmlImport.stem(stem)	The Stem into which data will be imported. Defaults to the ROOT stem.
XmlImport xmlImport.updateList(boolean)	XML contains a flat list of Stems or Groups which may be updated. Missing Stems and Groups are not created
XmlImport xmlImport.userProperties(file) XmlImport xmlImport.arouperSession(arouperSession)	Properties file for extra settings for import Operate within a certain grouper session (defaults to root session)
XmlImport xmlImport.ignoreInternal(boolean)	Ignore internal attributes, including group and stem uuids.
void xmlImport.importFromFile(file)	Import from an XML file
void xmlImport.importFromString(string)	Import from an XML string
void xmlImport.importFromUrl(url)	Import XML from a URL

Examples:

gsh 1% new XmlImport().importFromFile(new File("c:/temp/export.xml"))

```
gsh 1% grouperSession = GrouperSession.start(SubjectFinder.findById("mchyzer"));
gsh 2% stem = StemFinder.findByName(grouperSession, "aStem");
gsh 3% new XmlImport().stem(stem).updateList(true).userProperties(new File("C:/temp/
some.props")).grouperSession(grouperSession).importFromUrl(new URL("http://whatever.xml"));
```

#### -or- (without chaining)

gsh 3% xmlImport = new XmlImport(); gsh 4% xmlImport.stem(stem); gsh 5% xmlImport.grouperSession(grouperSession); gsh 6% xmlImport.importFromFile(new File("c:/temp/export.xml"))

#### Transactions

Transactions facilitate all commands succeeding or failing together, and perhaps some level of repeatable reads of the DB (depending on the DB). If there is an open transaction and an exception is thrown in a command, GSH will shut down so that subsequent commands will not execute outside of a transaction.

Command	Description
help("transaction")	print help information
transactionStatus()	print the list of nested transactions

transactionStart(" <groupertransactiontype>")</groupertransactiontype>	start a transaction, or make sure one is already
	started
	Can use: "READONLY_OR_USE_EXISTING",
	"NONE", "READONLY_NEW",
	"READ_WRITE_OR_USE_EXISTING",
	"READ_WRITE_NEW"
transactionCommit(" <groupercommittype>")</groupercommittype>	commit a transaction
	Can use: "COMMIT_NOW",
	"COMMIT_IF_NEW_TRANSACTION
transactionRollback(" <grouperrollbacktype>")</grouperrollbacktype>	rollback a transaction
	Can use: "ROLLBACK_NOW",
	"ROLLBACK_IF_NEW_TRANSACTION
transactionEnd()	end a transaction
	Note if it was read/write, and not committed or
	rolled back, this will commit and end

### **GrouperShell Variables**

gsh has several variables that can be set to modify runtime behavior

Variable	Description
GSH_DEBUG	Stack traces will be printed upon failure if true
GSH_DEVEL	Summaries of returned objects are not automatically printed if true
GSH_TIMER	Prints time spent evaluating each command if true

#### Example:

gsh 4% GSH\_DEVEL = true gsh 5% subj = findSubject("SD00125") gsh 6% sess = GrouperSession.start(subj) gsh 7% member = MemberFinder.findBySubject(sess, subj) gsh 8% p(member.getGroups()) group: name='etc:sysadmingroup' displayName='Grouper Administration:SysAdmin Group' uuid='6f77fb36b466-481a-84a7-7af609f1ad09'

# Note: you cannot encrypt passwords with GSH since the passwords end up in the GSH history. To encrypt passwords, issue the command:

C:\mchyzer\isc\dev\grouper-qs-1.2.0\grouper>java -jar lib\morphString.jar Enter the location of morphString.properties: conf/morphString.properties Type the string to encrypt (note: pasting might echo it back): The encrypted string is: ca8al5be4ad0fb45c6flb3ca0cfd9c9e

Questions or comments? <sup>1</sup> Contact us.

**<u>GROUPER</u>: About FAQ Software Documentation Archives Contribute WG Contact</u>** 

### **Grouper UI Components**

This page last changed on Dec 06, 2007 by jbibbee@internet2.edu.

**GROUPER:** About FAQ Software Documentation Archives Contribute WG Contact

## **Overview**

This document is current as of the v1.2.0 release.



### **Custom Components**

grouper

grouper-ui

custom-grouper-ui (optional)

custom subject provider (optional)

The Grouper API distribution includes the binary files for the Subject API implementation, which includes a JDBC provider. Source code for the Grouper UI is maintained as a separate module in the Internet2 Middleware CVS repository. Sites implementing Grouper will generally want to re-brand (and perhaps heavily customise) the native Grouper UI (see <u>Customising the Grouper UI</u>). Depending on the identity management / person repository(s) in use, a site may also need to implement a custom Subject provider

### **Build Product**

The Grouper UI build script compiles and combines the custom components in order to create a single web application build product. This step is responsible for ensuring that all required libraries (JAR files) and configuration files are available on the web application class path, i.e., in the *lib* or *classes* directories.

### **Application Server**

Once built, the web application is then deployed to an application server. Currently, only Tomcat has been tested.

The Grouper UI does not require any container specific configuration to work.

### Databases

Grouper requires a relational database. The default is HSQLDB, however, in principle, any database for which there is a JDBC driver and for which is supported by Hibernate can be used.

The Subject API can be configured to work with multiple sources (through sources.xml). A JDBC provider is provided with the Subject API distribution (an LDAP provider will be made available in the future), however, sites can implement their own providers.

Each time the Grouper UI is built, the sources.xml and hibernate.properties file are copied from grouper/ conf to the web application classes directory. Database components can, through these files, be configured to be on the same machine or separate machines.

# The Grouper QuickStart Distribution

The default Grouper <u>QuickStart</u> configuration uses the same HSQLDB database as a Grouper repository and as a source for the JDBC provider - the only source configured. In addition, Tomcat and the HSQLDB database run on the same machine.

Questions or comments? 1 Contact us.

**<u>GROUPER</u>:** <u>About</u> <u>FAQ</u> <u>Software</u> <u>Documentation</u> <u>Archives</u> <u>Contribute</u> <u>WG</u> <u>Contact</u>

### **Grouper UI Development Environment**

This page last changed on Dec 06, 2007 by jbibbee@internet2.edu.

**GROUPER:** About FAQ Software Documentation Archives Contribute WG Contact

# **Grouper UI Development Environment**

### Introduction

This document is current as of release v1.2.0.

There are many ways to set up a development environment using a variety of open source and commercial tools and application servers. The environment described here is the one used to develop the Grouper UI - however, you are free to use whatever setup works best for you.

I am an active developer, so the directory layout and build scripts I use are designed to facilitate development as well as final deployment. We normally use Eclipse as a Java IDE, and so some choices I made are biased to *cope* with the way Eclipse works. - Gary Brown, UoB

### **Directory structure**

In order to verify the extensibility of the UI, I have developed the Grouper UI and a custom (University of Bristol) version in parallel, using the same environment. A minimal implementation only requires a Grouper API installation in addition to a Grouper UI installation, however, any real world implementation will have site specific components as well:

### Grouper UI Development at the University of Bristol, UK

Component	Description
grouper	The Grouper API
grouper-ui	The Grouper UI
uob-grouper-ui	Bristol customisations to the Grouper UI
i2mi-subject	Bristol implementation of the Subject interface.
	Sites may be able to use a generic source adapter
	provided with The Subject interface distribution e.g.
	an LDAP adapter

grouper and grouper-ui are separate modules in the I2MI CVS repository.

uob-grouper-ui and i2mi-subject are separate modules in the CVS repository at Bristol.

I have all the CVS checkout directories as subdirectories at the same level (to help Eclipse), though this is not an absolute requirement, i.e., :

GrouperComplete grouper grouper-ui\* uob-grouper-ui\* i2mi-subject

\*Both directories contain a subdirectory *webapp* which itself has a directory structure that is consistent with a web application (see Architecture document)..

During development I may need to debug source code from any of the projects. I may also want to make code changes in the appropriate CVS checkout areas\*. In my ideal development environment I would be able to edit any source files and instantly see changes in the web application. A typical build script for a web application might create the web application directory structure in a new *build* directory and then either copy or make into a WAR (web application archive file), and then deploy to a Servlet container e.g. Tomcat. Using this approach every change requires a build and potential restart of the web application. Admittedly Eclipse will allow an ant script to be called when source files are modified, however, this can be overkill for a simple change to a JSP.

\*I could edit JSP and other files in the *build* or *deploy* directory, however, I would then need to copy the changes back to CVS - something I may well forget to do.

### **Setting Up Eclipse**

In Eclipse I create one *project* and pull in the *java/src* and *lib* directories from each of the 4 projects listed above. I can then set a single output directory where compiled Java classes are placed whenever I save a Java source file. JSP and other *content* files are trickier since they are saved *in situ* and not compiled to a separate destination. Normally I will be working on *either* the core Grouper UI *or* on Bristol customisations . In the Grouper UI *build.properties* file I can elect to have the *webapp* directory of grouper-ui *or* uob-grouper-ui be the web application root (configured in Tomcat)\*. I manually configure Eclipse to compile Java classes to the appropriate *webapp/WEB-INF/classes* directory.

\*Actually, any directory can be configured to be the web application root - I always choose either of the ones indicated when developing.

Any changes I make to the *local* JSPs are immediately picked up by Tomcat, however, I would need to run an ant script to obtain changes from the other project i.e. *uob-grouper-ui* to *grouper-ui*. Most sites which are not involved in the development of the Grouper UI should set *<institution>-grouper-ui/webapp* to be their web application root. If working with *Tomcat* and the build.properties *deploy* properties are set, the build script will automatically install your webapp on Tomcat such that Tomcat reads files from your *work area*.

A disadvantage of this approach is that it *pollutes* the CVS checkout area for one module with those from another, and I may be tempted to edit a file in the wrong location (though hopefully they are in different subdirectories). Assuming that site-specific changes are always in distinct subdirectories then on Unix it may well be possible to set up symbolic links from grouper-ui to, say, uob-grouper-ui.

Some changes e.g. adding new JAR files, modifying resources, changing Struts / tiles configuration files will always require a build and a web application restart.

#### **The Ant Script**

The following targets are available:

>ant help

```
Buildfile: build.xml help:
[echo] Please ensure you have read the documentation -
[echo] and created a build.properties file based on the template provided
[echo] The following targets are available - type the appropriate name:
[echo] 1) default
[echo]
          Simply builds, without cleaning, to the default.webapp.folder
[echo] 2) nice
[echo] Attempts to stop the Tomcat webapp before building.
[echo]
           Attempts to start the webappp afterwards
[echo] 3) clean
          Always removes the webapp.class.folder. May remove the
[echo]
           webapp.folder if webapp.folder.cleanable=true
[echo]
          On Windows this may fail as Windows tends to lock files
[echo]
[echo] 4) niceclean
          Combination of nice and clean
[echo]
[echo] 5) dist
           Cleans and then builds to subfolder of dist.home
[echo]
[echo] 6) war
[echo]
          Does dist and then makes a WAR file
[echo] 7) resources
[echo]
          Does not compile Java classes but 'refreshes' resources in
[echo]
          webapp.class.folder
[echo] 8) niceres
[echo] Does not compile Java classes but 'refreshes' resources in
[echo]
           webapp.class.folder and restarts webapp
[echo] 9) help
[echo]
          Displays this menu
[echo] 10) endhelp
```

echo] Subsequent invocation of ant with no target will run
echo] 'default' rather than help
echo] 11) starthelp
echo] Subsequent invocation of ant with no target will run 'help'
echo] 12) html
echo] Generate Javadoc - you must have done a 'default' build previously
echo] 13) exit
echo] Exit this menu without executing another target
input] Make your choice (default)>

The *nice* targets will only work if you are using Tomcat and have configured the deploy properties in build.properties, and have installed catalina-ant.jar with Ant.

See Customising the Grouper UI: <u>Customising the Build Process</u> for details on how to customise the build process.

**Questions or comments? Contact us.** 

GROUPER: About FAQ Software Documentation Archives Contribute WG Contact

### **Grouper Web Services**

This page last changed on Jan 24, 2009 by tbarton@uchicago.edu.

**GROUPER:** About FAQ Software Documentation Archives Contribute WG Contact

# **Grouper Web Services as of v1.4.0**

#### Introduction

Grouper web services (grouper-ws) is a J2EE web application which exposes common Grouper business logic through SOAP and REST. See <u>FAQ</u>.

To deploy the services, download the warfile and configure the property files (e.g. subject sources, databases, logging, etc). Configure <u>authentication</u>.

Note: there is a command line and java API web service client called Grouper Client

To implement a web service client:

- 1. Understand the object model. All grouper-ws services are operations based on simple data structures. The structures support Strings, ints, arrays, and structure references.
  - a. <u>Core web service API</u>
  - b. Example structure (only "getters" and "setters" are applicable properties)
  - c. Each operation has many samples (authmated captures, versioned, and up to date). <u>Here is an example</u>
  - d. Most options has a sensible default (e.g. MemberFilter defaults to All members)
  - e. Lookup objects in various (consistent) ways. e.g. to delete a group, you can pass the name or uuid of the group.
- Decide if you are using SOAP or REST (this is real REST, not Axis HTTP/XML)

   Both SOAP and REST support the same API
- 3. Inside SOAP and REST, each operation has two levels of complexity, the normal one, and the Lite one.
  - a. Normal operation: can usually be batched (support a list of inputs, e.g. add multiple groups at once), supports complex inputs (arrays or structures)
  - b. Lite operation: supports only inputs of scalars (no structures, no arrays... only Strings, ints, etc). In REST this also means that the request can be sent via query string only
- 4. If SOAP:
  - a. Implement based on the WSDL
  - b. There is a sample Java client with sample calls
- 5. If REST:
  - a. Decide what format you want to send and receive data. grouper-ws supports <u>XHTML, XML,</u> <u>and JSON</u>, as well as query strings for input (in URL or message body)
  - b. There are many <u>samples</u>

#### .NET client development guide

#### PHP client development guide

### **Guidelines For Working With Grouper Web Services**

- 1. There is a bug we are tracking with Axis, where if you skip String params, it will mix up the params. So, if you are passing a param to a web service, make sure you pass empty strings for all null params before the param
- 2. Code clients with a mindset that the service might change in subtle ways. e.g. a result code might be added (check for success flag element, not success result code), an element might be added in a result object, another input element might be added to end of list, etc. Expect elements to be added in data
- 3. Make sure there is a property in the client of the URL and version for the service. The version of the service might change the URL (up to service deployer)...

### Operations

<u>addMember</u>: assign a member to a group

 If already a member, that is ok

- Accepts batches of members (non-Lite)
- Accepts flag to say that any members not in batch should be removed (e.g. replace list)
- <u>deleteMember</u>: unassign a member from a group
  - ° If not a member, that is ok
  - Accepts batches of members (non-Lite)
- <u>getMembers</u>: return the members (including subject data) in a group (from direct or indirect membership)
  - Will accept member filter (All, Effective, Immediate, Composite)
  - Accepts batches of groups (non-Lite)
- getMemberships: under construction
  - Will accept member filter (All, Effective, Immediate, Composite)
  - Accepts batches of subjects and groups (non-Lite)
- <u>hasMember</u>: see if a subject is a member of a group
  - Will return true or false
  - Accepts batches of subject ids or identifers (returns batches of true's / false's) (non-Lite)
  - Will accept member filter (All, Effective, etc)
  - Can query on field (permission)
- <u>getGroups</u>: list groups for a subject
  - Will accept member filter (All, Effective, etc)
  - Accepts batches of subjects (non-Lite)
- groupSave
  - ° Create / update a group
  - Accepts batches of groups (non-Lite)
- groupDelete
- Delete a group
  - Accepts batches of groups (non-Lite)
- getGrouperPrivileges
  - View privileges for a subject and (group or stem)
  - Can view all privileges for the subject and (group or stem) or a specific privilege
- <u>assignGrouperPrivileges</u>
  - $^\circ\,$  Add or remove a privilege for a subject and (group or stem)
  - Will not fail if the privilege is already assigned or revoked
- findGroups
  - Can query for groups based on name, uuid, parent stem, or a substring query
  - Can create complex queries with group match (AND, OR, MINUS) (non-Lite)
- <u>findStems</u>
  - ° Can query for stems based on name, uuid, parent stem, or a substring query
  - Can create complex queries with group match (AND, OR, MINUS) (non-Lite)
- stemSave
  - Create / update a stem
  - Accepts batches of stems (non-Lite)
- stemDelete
  - Delete a stem
  - Accepts batches (non-Lite)
- memberChangeSubject
  - Change the subject of a current member
  - Accepts batches (non-Lite)

### Features

- API
  - Batched operations (e.g. add 100 subjects to a group at once). There is a separate server-side max-in-batch param in the grouper-ws.properties.
  - Transaction support (if any fails in one batch request, rollback all in that single batch request)
- Authentication
  - Let container or web server handle
  - o PKI
  - o http-simple-auth
  - o Source IP address filtering (TODO)
  - Custom authenticator
  - WS-Security
    - o PKI
    - o Kerberos

- Proxying. The web service can execute operations based on an underlying user, not the
- authenticating user. Note the authenticating user must have appropriate permissions
- Error Handling
  - Error codes and error messages are sent in responses, as well as warnings. In batched mode, batches of response codes are returned. In REST, the http status code is used as well.
- Clients
  - Grouper will provide a quick start with Java, and it is up to users to create their own clients. The SOAP and REST are based on the HTTP documents, so any programming language will work
- Web Service Implementation
  - Apache Axis for SOAP, and home-grown for REST

#### **Quick start**

Checkout the appropriate projects under <u>grouper-ws</u>, read the <u>README.txt</u> in the grouper-ws/grouper-ws cvs directory

### **Build Script**

The build script for grouper-ws is pretty basic. Generally just do the default (dist). There is also an "ant grouper" target to build a new grouper jar, and "ant quick" to do everything but generate the Axis files (takes 3 minutes).

```
C:\mchyzer\isc\dev\grouper\grouper-ws>ant
Buildfile: build.xml
dist:
clean:
   [delete] Deleting directory C:\mchyzer\isc\dev\grouper\grouper-ws\build
    [mkdir] Created dir: C:\mchyzer\isc\dev\grouper\grouper-ws\build\grouper-ws
    [mkdir] Created dir: C:\mchyzer\isc\dev\grouper\grouper-ws\build\dist
    [mkdir] Created dir: C:\mchyzer\isc\dev\grouper\grouper-ws\build\dist\grouper-ws
compile:
    [javac] Compiling 10 source files to C:\mchyzer\isc\dev\grouper\grouper-ws\build\grouper-ws
    [javac] C:\mchyzer\isc\dev\grouper\grouper-ws\src\grouper-ws\edu\internet2\middleware\grouper
\webservices\GrouperSer
viceServlet.java:33: warning: [deprecation] getEPRForService(java.lang.String,java.lang.String) in
 org.apache.axis2.tran
sport.TransportListener has been deprecated
    [javac] public class GrouperServiceServlet extends AxisServlet {
    [javac]
    [javac] 1 warning
generate-aar:
    [mkdir] Created dir: C:\mchyzer\isc\dev\grouper\grouper-ws\build\dist\webservices\classes
   [delete] Deleting directory C:\mchyzer\isc\dev\grouper\grouper-ws\build\dist\webservices\classes
    [mkdir] Created dir: C:\mchyzer\isc\dev\grouper\grouper-ws\build\dist\webservices\classes
    [copy] Copying 13 files to C:\mchyzer\isc\dev\grouper\grouper-ws\build\dist\webservices
\classes
      [jar] Building jar: C:\mchyzer\isc\dev\grouper\grouper-ws\webapp\WEB-INF\services
\GrouperService.aar
      [jar] Building jar: C:\mchyzer\isc\dev\grouper\grouper-ws\build\dist\grouper-ws.jar
    [mkdir] Created dir: C:\mchyzer\isc\dev\grouper\grouper-ws\build\dist\grouper-ws\WEB-INF
\classes
    [mkdir] Created dir: C:\mchyzer\isc\dev\grouper\grouper-ws\build\dist\grouper-ws\WEB-INF\lib
     [copy] Copying 12 files to C:\mchyzer\isc\dev\grouper\grouper-ws\build\dist\grouper-ws\WEB-INF
\classes
     [copy] Copying 30 files to C:\mchyzer\isc\dev\grouper_ws\build\dist\grouper-ws\WEB-INF
\lib
     [copy] Copying 11 files to C:\mchyzer\isc\dev\grouper\grouper-ws\build\dist\grouper-ws
      [jar] Building jar: C:\mchyzer\isc\dev\grouper\grouper-ws\build\dist\grouper-ws.war
BUILD SUCCESSFUL
Total time: 22 seconds
```

The system cannot find the batch label specified - end

C:\mchyzer\isc\dev\grouper\grouper-ws>

Notice the generate-aar target. This is what makes the axis archive, which is all the classes needed for axis to determine the wsdl, along with the services.xml config file.

Axis is  $\sim$ 40 jars, though most of them are pretty axis specific. There is an ant target which will compress most of these into one jar (axisBundle.jar). Here is the ant help:

C:\mchyzer\isc\dev\grouper\grouper-ws>ant help Buildfile: build.xml help: [echo] Please ensure you have read the documentation -[echo] and created a build.properties file based on the template provided [echo] [echo] The following targets are available - type the appropriate name: [echo] [echo] 1) default (dist) Simply builds, without cleaning, to the webapp.folder [echo] [echo] 2) clean Clean the webapp folder, and classfiles, and build [echo] [echo] 3) generate-aar [echo] Make the axis archive, which is the classfiles and services.xml that axis needs. You need to do this i f you ever change anything that changes the wsdl. You can do this automatically in dist by setting a property in the bu ild.properties [echo] 4) generate-axis-bundle-jar Take all the bundlable axis jars (in lib/axis-bundle), unjar, and jar back up into [echo] one jar [echo] BUILD SUCCESSFUL Total time: 0 seconds

The system cannot find the batch label specified - end

C:\mchyzer\isc\dev\grouper\grouper-ws>

#### To do's (post 1.3.0)

- 1. investigate backwards compatibility with Axis... discuss options
- 2. add find subject service
- 3. test more
- 4. unit test
- build a client jar back into web services to unit test
- 6. make some params to test stuff... (junit to throw exceptions in the middle of tx?)
- 7. come up with formatter and code style and remove all warnings
- 8. add logging filter
- 9. fix javadoc warnings
- 10. look into axis 1.4, see if error fixed, see if samples/wsdl changes, see about enums
- 11. add move subject service
- 12. add metadata service
- 13. add getGroups with batched groupLookup input
- 14. add batched privilege service, and add more url options to REST
- 15. add back in memberships service
- 16. filter getMember by privileges (find member?)
- 17. in rest add GET starting points with links to resources
- 18. improve auto-toString methods in resultMessage
- 19. look at acegi
- 20. add ip source filtering to grouper

Contact us if you have additional comments or suggestions.

**<u>GROUPER</u>:** <u>About</u> <u>FAQ</u> <u>Software</u> <u>Documentation</u> <u>Archives</u> <u>Contribute</u> <u>WG</u> <u>Contact</u>
## **Authentication for Grouper Web Services**

This page last changed on Sep 02, 2008 by sanjay.vivek@ncl.ac.uk.

Contact us if you have additional comments or suggestions.

**<u>GROUPER:</u>** <u>About</u> <u>FAQ</u> <u>Software</u> <u>Documentation</u> <u>Archives</u> <u>Contribute</u> <u>WG</u> <u>Contact</u>

## Authentication for Grouper Web Services as of v1.3.0

#### **Default authentication**

Out of the box, grouper-ws uses container authentication (non-rampart). The web.xml protects all services and expects the users to be in the role "grouper\_user". For tomcat, in the tomcat-users.xml, just have entries like this, and you are all set:

```
<role rolename="grouper_user"/>
<user username="jota" password="whatever" roles="grouper_user"/>
<user username="jobr" password="whatever" roles="grouper_user"/>
<user username="eldo" password="whatever" roles="grouper_user"/>
```

Note that users to the web service need to be Subjects, and you can configure the default source in the grouper-ws.properties especially if you have subjectId overlap in various sources.

Note the default authentication in grouper-ws is http-basic, so for this and other reasons make sure your deployments of grouper-ws are protected with SSL.

Note that, for some container technologies, container authentication can be externalized in various ways. A common deployment configuration is to externalize tomcat authentication to Apache 2.2+ using the AJP protocol. This permits several popular authentication technologies to be used in conjunction with grouperws.

#### **Custom authentication plugin**

If you want custom authentication (e.g. pass in a token, and decode it), then implement the interface edu.internet2.middleware.grouper.ws.security.WsCustomAuthentication and configure your fully qualified classname in the grouper-ws.properties. The default is an implementation of this interface as an example: edu.internet2.middleware.grouper.ws.security.WsGrouperDefaultAuthentication, which just gets the user from the container: httpServletRequest.getUserPrincipal().getName()

#### Rampart

Rampart is Jakarta's WS-Security implementation. We have vanilla <u>Rampart authentication</u> working with grouper-ws (thanks to Sanjay Vivek). Unfortunately it doesn't work out of the box since it seems Rampart and basic auth cannot work together in the web app. If you want to run basic auth and rampart at the same time, you should deploy two separate web apps.

Note the URL for rampart in grouper-ws is the same, it will look like this: /grouper-ws/services/ GrouperService

Also, for Rampart, you need custom logic to authenticate users. To use rampart, configure the grouper-ws.properties entry: ws.security.rampart.authentication.class. An example is: edu.internet2.middleware.grouper.ws.security.GrouperWssecSample. Until you configure that, clients will get a 404 http status code. This assumes you are using WSPasswordCallback, if not, just provide your own class directly to the services.xml file (and grouper-ws requires you have an implementation of the interface anyway which wont be executed).

You need to tell grouper that wssec is enabled in the web.xml servlet param (uncomment):

```
<servlet>
        <servlet-name>AxisServlet</servlet-name>
        <display-name>Apache-Axis Servlet</display-name>
        <servlet-class>edu.internet2.middleware.grouper.ws.GrouperServiceAxisServlet</servlet-
class>
        <load-on-startup>1</load-on-startup>
```

```
Document generated by Confluence on Feb 02, 2009 16:50
```

Also you need to comment out the container auth in web.xml:

Then you need to enable the correct .aar file.

- If you are using a binary grouper-ws.war, just rename the following two files
  - /WEB-INF/services/GrouperService.aar to /WEB-INF/services/GrouperService.aar.ondeck
     /WEB-INF/services/GrouperServiceWssec.aar.ondeck to /WEB-INF/services/ GrouperServiceWssec.aar
- If you are building, just set the param in the build.properties: webapp.authentication.use.rampart Here is a sample client

#### HTTP basic authentication (use)

In the web.xml for the grouper-ws project, protect all services:

```
<security-constraint>
                <web-resource-collection>
                        <web-resource-name>Web services</web-resource-name>
                        <url-pattern>/services/*</url-pattern>
                </web-resource-collection>
                <auth-constraint>
                        <!-- NOTE: This role is not present in the default users file -->
                        <role-name>grouper_user</role-name>
                </auth-constraint>
        </security-constraint>
        <!-- Define the Login Configuration for this Application -->
        <login-config>
                <auth-method>BASIC</auth-method>
                <realm-name>Grouper Application</realm-name>
        </login-config>
        <!-- Security roles referenced by this web application -->
        <security-role>
                <description>
                        The role that is required to log in to the Manager
                        Application
                </description>
                <role-name>grouper_user</role-name>
        </security-role>
</web-app>
```

Now send the user and pass in the web service client. Here is an example with Axis generated clients:

```
auth.setPassword("pass");
```

options.setProperty(HTTPConstants.AUTHENTICATE, auth);

#### Here is an example with a manual HttpClient:

```
HttpClient httpClient = new HttpClient();
       GetMethod getMethod = new GetMethod(
                "http://localhost:8091/grouper-ws/services/GrouperService/addMemberSimple?
groupName=aStem:aGroup&subjectId=10021368&actAsSubjectId=GrouperSystem");
       httpClient.getParams().setAuthenticationPreemptive(true);
        Credentials defaultcreds = new UsernamePasswordCredentials("user", "pass");
       httpClient.getState().setCredentials(new AuthScope("localhost", 8091), defaultcreds);
```

httpClient.executeMethod(getMethod);

#### ActAs configuration

To enable web service users to act as another user (proxy), enable the setting in the grouper-ws grouper.properties

# Web service users who are in the following group can use the actAs field to act as someone else ws.act.as.group = aStem:aGroup

If you specify a group name in there, you can pass in the actAs field if you connect to the web service as a user who is in the ws.act.as.group group. Here is an example with the axis generated client.

```
//set the act as id
       WsSubjectLookup actAsSubject = WsSubjectLookup.class.newInstance();
        actAsSubject.setSubjectId("GrouperSystem");
        addMember.setActAsSubjectLookup(actAsSubject);
```

There are advanced settings, you can specify multiple groups in the grouper-ws.properties, and you can even limit who the users can act as (in a specific group).

#### Additional Information

The G-FIV-O Project has produced a report that delves further into options for securing Grouper Web Services.

GFIVO: Multiple Security Mechanisms Web Services Deployment

## **Grouper Web Services FAQ**

This page last changed on Apr 17, 2008 by tbarton@uchicago.edu.

· Can I run grouper web services against any version of grouper?

No, you should use the version of grouper bundled in the web services. This makes it a little complicated if you have a UI / GSH / etc running against your grouper DB. You will need to keep them in sync...

• Why are there three flavors or web services (SOAP, XML-HTTP, REST-Lite)?

SOAP is supported since many schools required it. REST-Lite is supported because many schools required it. XML-HTTP is supported because Axis2 gives it for free when building a SOAP web service. This way the same SOAP interface can be accessed by clients who only want to talk HTTP and XML and not SOAP.

• Why is the rampart .aar file named GrouperServiceWssec.aar, but the URL is still /services/ GrouperService?

The URL for ramprt or not is the same. Inside the services.xml in the .aar files, it configures the app name. grouper-ws will not work if you change this (unless to do other build activities also)

• Why is there a "simple" operation for every non simple operation in SOAP and XML-HTTP?

The non-simple operations are batchable if the client wants to do one operation multiple times with one request. The simple operations are for if the client only needs to do one thing (e.g. assign a member to a group and not assign multiple members to a group). Also, there is a valuable side effect that for the XML-HTTP, if the operation only has scaler input params (and not complex types or arrays), that the input does not need XML, it can be in the query string for GET or in the http form param pairs in the body for POST.

It is confusing that there are so many different ways to call grouper via web service. The documentation will be improved to make it easier to find the best strategy.

• Why element named "return" (by axis)

This is an unfortunate "feature" by axis, the default element is named "return" which is a keyword in many programming languages, so if the language automatically converts the xml to an object graph, then it will be broken. Chris will followup with Axis to see if there is a fix for this

• Why returning error codes and messages and not just use SOAP faults?

For two of three of the flavors of web services SOAP faults are not an option since they are not SOAP. Also, for SOAP batched, the status of each line item needs to be returned for the client to process, and a SOAP fault would preclude that. Also, the fewer SOAP specific features that are used, the more widespread the compatibility will be. All three flavors of web service use the same underlying logic, so the more consistent the better.

· Can we add a service for subject search?

Yes, this will be added

## **Import-Export**

This page last changed on Jan 08, 2009 by tzeller@memphis.edu.

GROUPER: About FAQ Software Documentation Archives Contribute WG Contact

## XML Import / Export for Grouper v1.4.0

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

Grouper v1.2.0+ 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

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).

The XML formats for import and export are very similar, however, there are some differences.

The export format:

- defines what is actually exported,
- · includes some meta data about the export,

while the import format:

- · allows import options to be embedded in the XML,
- defines additional attributes for Stems and Groups which may affect the importing of Stems and Groups,
- does not require all of the information that is exported.

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. Since v1.3.0 system attributes are maintained by default, which is the desired behavior if migrating a registry, however, this can cause a problem if you want to copy part of the registry by exporting it and importing it into a new stem because the uuids of imported groups and stems already exist. v1.4.0 introduces a new command line argument *-ignoreInternal* (see below) which ensures that uuids and other internal attributes are ignored.

## **Export Tool in More Detail**

A Java class, XmlExporter, provides the export functionality. It can be run from the command line, from within Java code, or using gsh:

bin/gsh.sh -xmlexport <command line arguments>

The command line usage is:

Command	and Summary of args.	
args: -h	Prints this message	
args:	subjectIdentifier [(-id] [-name )] [-relative] [-	
	childrenOnly] [-includeParent] fileName [properties]	

The above export args. can be explained as follows:

Command	Description
subjectIdentifier	Identifies a Subject 'who' will create a
	GrouperSession.
-id	The Uuid of a Group or Stem to export. Defaults to
	the ROOT stem.
-name	The name of a Group or Stem to export. Defaults to
	the ROOT stem.
-relative	If id or name specified do not export parent Stems.
-includeParent	If id or name identifies a Group and -relative is
	selected, export the Group and its parent Stem.
-childrenOnly	If id or name identifies a Stem and -relative is
	selected export child Stems and Groups, but not the
	stem itself.
filename	The file where exported data will be written. Will
	overwrite existing files.
properties	The name of an optional Java properties file.
	Values specified in this properties file will override
	the default export behavior documented in the
	XmlExporter javadoc.

The JavaDoc describes the export methods, including a method which can be used to export an arbitrary Collection of Stems, Groups, Subjects or Memberships returned by various Grouper API methods. This means that the results of any *list* or *search* methods can be exported.

An XML Schema which describes the exported XML is available here.

If a relative export is performed, the export tool treats group members, list members or privilegees which are groups, and which are *descendants* of the export stem in a special manner. The Subject Identifier, which, for groups, is usually the group name, is modified so that the export stem name is replaced by an asterix, thus, if performing a relative export of uob:artf, a reference to the staff group would become \*staff rather than uob:artf:staff. The import tool will replace the asterix with the import stem name. In this way the relationship between groups can be maintained.

Examples of exported data are available here.

## **Import Tool in More Detail**

A Java class, XmlImporter, provides the import functionality. It can be run from the command line, from within Java code, or using gsh:

bin/gsh.sh -xmlimport <command line arguments>

The command line usage is:

Com	mand	Summary of args.	
args: -h		Prints this message	
args:		<pre>subjectIdentifier [(-id -name -list)] [-ignoreInternal]</pre>	
		[-noprompt] filename [properties]	

The above import args. can be explained as follows:

Commands	Description
subjectIdentifier	Identifies a Subject 'who' will create a
	GrouperSession.
-id	The Uuid of a Stem into which data will be imported.
	Defaults to the ROOT stem.

-name	The name of a Stem into which data will be imported. Defaults to the POOT stem
-list	File contains a flat list of Stems or Groups which may be updated. Missing Stems and Groups are not created.
-ignoreInternal	Do not attempt to import internal attributes including Group/Stem uuids. Overrides property: import.data.ignore-internal-attributes-and-uuids
-noprompt	Do not prompt user to confirm the database that will be updated
filename	the file to import
properties	The name of an optional Java properties file.
	Values specified in this properties file will override
	the default import behavior documented in the XmlImporter javadoc.

The JavaDoc describes the load methods.

An XML Schema which describes the format of XML which can be loaded is available here.

It is possible to generate an XML file which validates against the schema, but which does not load properly. The annotations in the schema describe appropriate usage of attributes and elements.

The Grouper QuickStart includes a demo registry. <u>quickstart.xml</u> is a minimal XML import file which creates the demo registry\*.

When generating XML in the import format, it is likely that relationships between stems and groups will need to be specified. This is problematic because the uuids of groups and stems are unknown prior to creation. In addition, it is not always possible to know the full name of a new Stem or Group, as this will depend on which stem it is imported into. When importing Subjects that are groups, the import tool examines the identifier attribute and makes any necessary changes before further processing. The following notations are recognised:

Notation	Description
*SELF*	Refers to the context group for which the Subject
	is being processed as a member*, list member or privilegee.
	*Actually the API will prevent a Group becoming a member of itself
*	As desribed above, * is replaced with the name of the Stem where the XML is to be imported
.:	Replace with the name of the Stem which contains the context group.
:	Replace with the parent Stem of the Stem which contains the context group. May occur multiple times.

## Notes from the Field

Some of the example xml and the xsd referenced above are inconsistent with the v1.2.0+ implementation of the xml import/export tool. Here are some details you need to know to successfully load members into groups using the xml import method.

1. The <subject> element requires the 'immediate' attribute. Best practice is to fully reference each subject, giving its source, type, and declaring it to be an immediate membership. So, instead of

<list field="members"> <subject id="someId"/> </list>

use

t field="members"> <subject id="someId" source='someSource' type='person' immediate='true' / > </list></list>

Questions or comments? <sup>1</sup> Contact us.

**GROUPER:** About FAQ Software Documentation Archives Contribute WG Contact

## **Initializing Administration of Privileges**

This page last changed on Jan 04, 2009 by tbarton@uchicago.edu.

**<u>GROUPER</u>:** <u>About</u> <u>FAQ</u> <u>Software</u> <u>Documentation</u> <u>Archives</u> <u>Contribute</u> <u>WG</u> <u>Contact</u>

## Initializing Administration of Grouper Privileges as of v1.4.0

GrouperSystem is the root-like principal used to manage assignment of privileges in Grouper. In addition to GrouperSystem, externally authenticated members of the <u>wheel group</u> can choose when to act with root-like privileges.

If you've enabled the wheel group, you must create it and add members. GrouperShell acts as GrouperSystem and can bootstrap the necessary naming stem(s), group, and memberships.

The wheel group is available as of Grouper v1.0.

#### **Enabling the Wheel Group**

The wheel group is enabled and named in conf/grouper.properties :

#### conf/grouper.properties

```
# A wheel group allows you to enable non-GrouperSystem subjects to act
# like a root user when interacting with the registry.
groups.wheel.use = true
```

# Set to the name of the group you want to treat as the wheel group. # The members of this group will be treated as root-like users. groups.wheel.group = etc:sysadmingroup

#### Automatically Creating the Wheel Group

To automatically create the wheel group :

#### conf/grouper.properties

configuration.autocreate.system.groups = true

#### Using GrouperShell to Create the Wheel Group

To create the wheel group using GrouperShell :

#### GrouperShell

```
gsh 0% addRootStem("etc", "Grouper Administration")
stem: name='etc' displayName='Grouper Administration' uuid='f7687876-2c94-4635-997c-f2793fb8152d'
gsh 1% addGroup("etc", "sysadmingroup", "SysAdmin Group")
group: name='etc:sysadmingroup' displayName='Grouper Administration:SysAdmin Group' uuid='6f77fb36-
b466-481a-84a7-7af609f1ad09
```

#### Adding Members to the Wheel Group

Whether you've set the wheel group to be automatially created, or you've used GrouperShell to create it, you'll need to add members to the wheel group using GrouperShell :

#### GrouperShell

```
gsh 0% addMember("etc:sysadmingroup", "SD00125")
true
```

In this example "SD00125" is the subjectId of a person, as determined outside of gsh by, in this case, an LDAP query to a directory that acts as a subject source to Grouper:

% ldapsearch \-b dc=kitn,dc=edu uid=tbarton dn: kitnEduPersonRegId=SD00125,ou=people,dc=kitn,dc=edu objectClass: top objectClass: person objectClass: inetOrgPerson objectClass: kitnEduPerson kitnEduPersonRegId: SD00125 cn: Barton, Tom sn: Barton description: Professor, Mathematics uid: tbarton

Questions or comments? <sup>1</sup> Contact us.

GROUPER: About FAQ Software Documentation Archives Contribute WG Contact

## Intro FAQ

This page last changed on May 28, 2008 by tbarton@uchicago.edu.

GROUPER: About FAQ Software Documentation Contribute WG Contact

## Intro FAQ about Grouper

Grouper is Licensed under the Apache 2.0 license. For technical FAQ, see the Technical FAQ.

- 1. What is Grouper?
- 2. Who needs Grouper?
- 3. <u>How can Grouper help my institution</u>?
- 4. How does Grouper differ from other solutions?
- 5. What do I need to use Grouper?
- 6. Does Grouper integrate well with others?

#### 1. What is Grouper?

•••

2. Who needs Grouper?

•••

3. How can Grouper help my institution?

•••

#### 4. How does Grouper differ from other solutions?

...

#### 5. Who developed Grouper?

•••

#### 6. Does Grouper integrate well with others?

•••

Questions or comments? <sup>1</sup> Contact us.

GROUPER: About FAQ Software Documentation Contribute WG Contact

## License

This page last changed on Jul 18, 2007 by jbibbee@internet2.edu.

GROUPER: About FAQ Software Documentation Contribute WG Contact

Note: Information regarding the **Grouper license** may be found here: <u>http://</u><u>middleware.internet2.edu/dir/groups/grouper/license.html</u>

Questions or comments? <sup>1</sup> Contact us.

## **Prerequisites**

This page last changed on Jan 04, 2009 by tbarton@uchicago.edu.

**<u>GROUPER</u>: About FAQ Software Documentation Archives Contribute WG Contact</u>** 

## Prerequisites as of v1.4.0

The Grouper API requires:

- Java
- database (e.g. MySQL, Oracle, PostgreSQL, etc.)

The Grouper UI and WS packages also require:

- <u>Ant</u>
- servlet container (e.g. <u>Tomcat</u>)
- web server (e.g. <u>Apache</u>)

The Specsheet provides additional details regarding requirement specifications.

The **Project Layout** describes the directory structure of Grouper packages.

## **Relational Database**

Grouper uses Hibernate to persist objects in a relational database, called the **Groups Registry**. Hibernate in turn uses JDBC for database connectivity. The .jar file containing the JDBC driver for the RDBMS of your choice must be available during installation.

All of Grouper's access to the underlying database is by means of a single account. The username and password or other authentication token for this account must also be available during installation.

The Grouper distribution includes the free and open source HSQLDB relational database, which is used in conjunction with testing the compiled code.

## **Web Server**

Although it is possible to run Grouper without a web server, it is likely needed for a production deployment. The web server will restrict access to the Grouper application, authenticate your users, optionally authenticate the special GrouperSystem account, and implement SSL.

If you will use <u>Apache v2.2+</u>, configure apache to load mod\_proxy and mod\_proxy\_ajp and include configuration directives similar to the following:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
ProxyRequests Off
```

Then, in tomcat's server.xml, ensure that the Connector element for port 8009 is uncommented and contains directives similar to these:

```
<Connector port="8009"
enableLookups="false" redirectPort="8443" protocol="AJP/1.3"
tomcatAuthentication="false"/>
```

If you will use <u>Apache v1.3 or v2.0.X</u>, configure apache to load the <u>mod\_jk connector</u>. The following configuration directs Apache to use mod\_jk to redirect queries for Grouper to Tomcat. This may be done by including the following text directly in httpd.conf, or making a separate file and including it in httpd.conf.

```
<IfModule \!mod_jk.c>
LoadModule jk_module libexec/mod_jk.so
</IfModule>
JkWorkersFile "/usr/local/tomcat/conf/jk/workers.properties"
JkLogFile "/usr/local/apache/logs/mod_jk.log"
JkLogLevel emerg JkMount /grouper/* ajp13
```

- Add address="127.0.0.1" to Tomcat's server.xml inside the <Ajp13Connector> configuration element to prevent off-host access.
  - For Tomcat 5.5 or newer, add request.tomcatAuthentication="false" to the <Ajp13Connector> configuration element in server.xml to ensure that the user's identity is passed from Apache to the servlet environment.
  - For Tomcat 5.0.x or older, add tomcatAuthentication="false" to the <Ajp13Connector> configuration element in server.xml to ensure that the user's identity is passed from Apache to the servlet environment.
  - *Tomcat 4.1.x* defaults to having the Coyote connector enabled in /conf/server.xml. This fails with mod\_jk and must be commented out. Then, uncomment and modify the traditional AJP 1.3 connector as indicated above.
- The AJP13Connector for tomcat is not compatible with the new JMX support. To remove some warnings that will appear in the Tomcat log every time Tomcat is restarted, comment out all of the JMX stuff (anything that says "mbeans") from server.xml.

#### Apache-based User Authentication

The interaction between the Grouper UI and an Apache-based local authentication system is implemented by providing the UI with the identity of the browser user through REMOTE\_USER. Any authentication system that is capable of protecting a block of webspace using httpd.conf and populating the REMOTE\_USER header variable is compatible with Grouper. This associates the appropriate authentication mechanism with the URL of the Grouper servlet, ensuring users authenticate and that their login name is passed to Grouper. The following example demonstrates use of a very basic authentication method with the Grouper UI:

```
<Location /grouper>
SSLRequireSSL
AuthType Basic
AuthName "ExampleU Login Service"
require valid-user
ProxyPass ajp://localhost:8009/grouper/
</Location>
```

Note that the ProxyPass declaration is included only when using mod\_proxy\_ajp.

#### Grouper UI-integrated User Authentication

The Grouper UI optionally supports direct integration of external authentication systems with the UI servlet. If this style of providing external authentication services to Grouper is chosen, REMOTE\_USER and use of Apache-based user authentication is not needed. See <u>How to Customize Authentication in the Grouper UI</u>.

#### **Grouper Web Service Authentication**

Grouper Web Services can be protected by

- REMOTE\_USER provided by the servlet container it runs in,
- Apache Rampart for WS-Security style of access protection, or
- a custom authentication plug-in.

See Authentication for Grouper Web Services for details.

## **Project Layout**

The Grouper API, UI, and WS packages must be unpacked under the same parent directory in order to install successfully. The parent directory must be writable because the Grouper UI build process will attempt to create a build directory under the parent directory.

The directory structure of the unpacked distributions is:

top level of API package
Configuration files for Grouper and third party
components
Grouper command line utilities
Compiled code, javadoc, and other generated
artifacts
Third party .jar files required by the Grouper API
Default location for log files
Currently holds stub instructions for the binary
distribution builder
Java source for Grouper API and API test suite
License under which Grouper may be used
Ant configuration file
top level of UI package
Contributed software
Grouper UI documentation
Java source for Grouper UI
UI properties files and other resources
Directory containing the as-built and customized UI
servlet
Images and styles for the UI
Generic styles
Taglibs and other structural definitions
top level of Web Services package
Grouper WS servlet
Compiled code
Grouper WS documentation
Third party .jar files required by Grouper WS
Configuration files
Servlet files
Apache .aar files
Sample Axis-generated client
Sample "Lite" client
used for testing

**Note**: The file grouper/lib/README lists the third party software used by Grouper and identifies the version, source, and license for each.

Questions or comments? O <u>Contact us</u>.

**<u>GROUPER</u>: About FAQ Software Documentation Archives Contribute WG Contact</u>** 

## **Software Download**

This page last changed on Jan 05, 2009 by tbarton@uchicago.edu.

**GROUPER:** About FAQ Software Documentation Archives Contribute WG Contact

## **Download Grouper**

#### Grouper v1.4.0

Software release: [4-January-2009]

This is the most recent stable version.

Release component	Description
Grouper QuickStart v1.4.0	Designed to get a working demo up and running quickly. Contains the Grouper components, a demo database, a database engine, and setup instructions. Just add Tomcat and stir
Grouper API Binary v1.4.0	Binary release of the Grouper Application Program Interface and associated utilities. Cf. <u>Getting started</u> with the Grouper API Binary Distribution
Grouper API Source v1.4.0	Contains the full source for the Grouper Application Program Interface.
Grouper UI v1.4.0	Contains the full source for the Grouper User Interface.
Grouper WS v1.4.0	Contains the full source for the Grouper Web Services Interface.
Grouper Client v1.4.0	Experimental client for Grouper LDAP and Web Services, in binary form. <u>Grouper Client v1.4.0</u> source is also available.
Contributed Software	Provided by the Grouper community, in addition to the above core Grouper products.

v1.4.0 Release Notes - Highlights of the v1.4.0 release, upgrade instructions, and detailed log of changes.

#### **Older versions**

<u>Archives</u> - For a complete listing of feature updates and changes to Grouper across previous releases, including archived documentation.

For an explanation of Grouper specific terms and definitions, see the <u>Glossary</u>. For detailed information regarding the deployment of Grouper, refer to the System Administration section of the main <u>Grouper</u> <u>Product</u> Documentation page.

#### If You're Interested...

Licensed under the Apache 2.0 license. <u>Specsheet</u> offers operational specifications for the development and deployment of Grouper. <u>Grouper Working Group</u> information relating to the Grouper project and software development can be found here. Or go to the <u>GrouperWG</u>wiki.

<u>CVS</u> - Access the Grouper source code via the web:

- Connection type: pserver
- User: anoncvs
- Passwd: <your email address>
- Host: anoncvs.internet2.edu
- Repository Path: /home/cvs/i2mi
- <u>Use default port</u>: yes

Access the complete Grouper source code via the command line:

cvs -z3 -d :pserver:anoncvs@anoncvs.internet2.edu:/home/cvs/i2mi login

cvs -z3 -d :pserver:anoncvs@anoncvs.internet2.edu:/home/cvs/i2mi co -r GROUPER\_1\_3\_1 grouper cvs -z3 -d :pserver:anoncvs@anoncvs.internet2.edu:/home/cvs/i2mi co -r GROUPER\_UI\_1\_3\_1 grouper-ui cvs -z3 -d :pserver:anoncvs@anoncvs.internet2.edu:/home/cvs/i2mi co -r GROUPER\_WS\_1\_3\_1 grouper-ws

Bug Reports - Bugs submitted and fixes found here. Note: You will need to create a login with Jira.

**NOTE WELL**: All Internet2 Activities are governed by the <u>Internet2 Intellectual Property Framework</u>.

#### Contact

🕜 If you have a question about the Grouper software, 🟮 <u>Contact us</u>.

**<u>GROUPER</u>:** <u>About</u> <u>FAQ</u> <u>Software</u> <u>Documentation</u> <u>Archives</u> <u>Contribute</u> <u>WG</u> <u>Contact</u>

## Grouper change log v1.4

This page last changed on Jan 27, 2009 by mchyzer.

This document lists instructions for people with existing groups installations on how to upgrade to newer versions of grouper (or grouper related products). If you notice something missing please let us know.

The instructions are in descending order based on date/release. You will find instructions below for Grouper, Grouper-ws, Grouper-ui, GSH, USDU, etc. It is assumed if you are running grouper-ui that you will perform both the grouper upgrade notes, and the grouper-ui upgrade notes. It is understood that you will get the new source/javadoc/etc files, this document addresses configurations, jars, etc.

#### Grouper

- 2009/1/25: v1.4.1 GROUPER\_1\_4\_BRANCH: the stem name index was changed to unique from non-unique. Note, if you have problems adding this index, you need to remove stems with duplicate names from the registry. Obviously this needs to be done with care, and you might want to backup before attempting, or contact grouper-dev for help
  - Run: grouper\_home/bin/gsh -registry -check
    - Then look at the result script, and execute that script, perhaps with:
      - grouper\_home/bin/gsh -registry -runsqlfile C:/mchyzer/isc/dev/grouper\_v1\_4/grouper/ ddlScripts/grouperDdl\_20090125\_08\_39\_44\_148.sql
  - or- you can do this manually, change the stem name index to unique, and update grouper\_ddl table Grouper entry DB\_VERSION to 13
  - $^\circ\,$  -or- (not recommended) you could export the registry to backup, rebuild the registry with run:
    - gsh -xmlexport GrouperSystem /whatever/20090125\_1\_4.xml
    - grouper\_home/bin/gsh -registry -drop -runscript
    - gsh -xmlimport GrouperSystem /whatever/20090125\_1\_4.xml
- 2009/1/25: v1.4.1 GROUPER\_1\_4\_BRANCH: merge grouper.properties with grouper.example.properties, adding properties: junit.test.loader, junit.test.ldappc, junit.test.ddl
- 2008/11/14: v1.4 HEAD: compare and merge all config files with examples, things were rearranged, organized, etc
- 2008/10/29: v1.4 HEAD: update morphString.jar, algorithm changed
- 2008/10/14: v1.4 HEAD: update jars for config checking:
  - delete lib/grouper/commons-discovery-0.4.jar
  - delete lib/grouper/jamon-2.7.jar
  - update invoker.jar
  - add commons-discovery.jar
  - add jamon.jar
  - update morphString.jar
  - update p6spy.jar
  - update subject.jar : note, if you have a custom source, you need to implement some new methods and recompile
- 2008/09/30: v1.4 HEAD: all jars have their version number removed, might want to delete old, and get new. e.g. from quartz-1.6.0.jar to quartz.jar
- 2008/09/23: v1.4 HEAD: <u>numParameters is now optional in sources.xml</u>. You can remove any numParameters params in sources.xml: <pre
- 2008/09/15: v1.4 HEAD: Subject API converted to use C3P0 pooling instead of DBCP.
  - delete: lib/commons-dbcp.jar
  - <sup>°</sup> delete: lib/commons-pool.jar
  - update: lib/commons-logging.jar
  - delete: lib/subject-0.3.0-rc1-cvs.jar
  - add: lib/subject.jar
  - compare the sources.xml with the sources.example.xml, and add the elements and comments about jdbcConnectionProvider. You can leave the value blank for C3P0, or fill it in. If you are using the same credentials for grouper and a subject source, remove the credentials from sources.xml and put edu.internet2.middleware.grouper.subj.GrouperJdbcConnectionProvider as the jdbcConnectionProvider
- 2008/09/14: v1.4 HEAD: <u>Passwords encrypted in external files</u>. Add the jar: morphString.jar. Add morphString.example.properties, and morphString.properties. In sources.xml, replace the sourceadapter classes
  - FROM: edu.internet2.middleware.subject.provider.JDBCSourceAdapter TO: edu.internet2.middleware.grouper.subj.GrouperJdbcSourceAdapter
  - FROM: edu.internet2.middleware.subject.provider.JNDISourceAdapter TO: edu.internet2.middleware.grouper.subj.GrouperJndiSourceAdapter

If you want to take advantage of external encrypted passwords (optional):

- In morphString.properties, set the encrypt.key entry to a random alphanumeric string, or a pathname of a file containing the alphanumeric string

- In sources.xml, and grouper.hibernate.properties, encrypt the passwords with: java -jar morphString.jar, and put results in a file, and put the file path where the passwords were in sources.xml or grouper.hibernate.properties

• 2008/08/14: v1.4 HEAD: <u>UUID/ID cols reduced</u>, and <u>Field foreign key is now field\_id</u>. Delete any check constraints you have in your database. E.g. on the grouper\_fields table. Start grouper to see in the logs a message that says to run a script, or do an: ant schemaexport. You should review the script before running it, as sometimes ddlutils does weird things. Also, in prod you should export your data before running. Once you are comfortable with the result after running the script and checking your grouper data, set grouper.properties ddlutils.dropBackupUuidCols to true, and ddlutils.dropBackupFieldNameTypeCols to true, then: ant schemaexport, to get the script to drop the backup uuid/id cols. I would run all db scripts in prod with a DB ide (e.g. toad) or manually so that you can address any issues (see link for example issues)

2008/07/27: v1.4 HEAD: DDL management changed.

- Copy the DDL section from the grouper.example.properties into your grouper.properties, configure them

- Add the jar: ant-1.7.1.jar

- run: ant schemaexport to sync everything up (or generate the script to sync everything up, depending on grouper.properties)

- Remove any schema-export.sql files
- 2008/07/22: v1.4 HEAD: If you were already using grouper loader, rename the table grouploader\_log to grouper\_loader\_log. If you already had the grouper\_ddl table, delete the java\_version column
- 2008/07/21: v1.4 HEAD: Add these jars: quartz-1.6.0.jar, commons-cli-1.1.jar, bsh-2.0b4.jar. GSH, loader, and usdu are in grouper now, with start scripts in grouper\_home/bin. See this jira report
- 2008/07/21: v1.4 HEAD: The java package structure was refactored. If you have code that doesnt compile, organize your imports. See this jira report
- 2008/07/20: v1.4 HEAD: Compare the grouper.properties with the grouper.example.properties, and add the section regarding group attribute validation via regex
- 2008/07/09: v1.4 HEAD: Everyone needs to tweak your log4j.properties:

Change FROM (remove .ErrorLog and .DebugLog):

log4j.logger.edu.internet2.middleware.grouper.ErrorLog	= ERROR, grouper_error
#log4j.logger.edu.internet2.middleware.grouper.DebugLog	= INFO, grouper_debug
TO:	

log4j.logger.edu.internet2.middleware.grouper	= ERROR, grouper_error
#log4j.logger.edu.internet2.middleware.grouper	= INFO, grouper_debug

and REMOVE these lines entirely:

## Grouper Test Logging

log4j.logger.edu.internet2.middleware.grouper.TestLog = INFO, grouper\_stdout

(probably doesn't affect you, but) If you have code that used the classes ErrorLog, DebugLog, or TestLog, these are removed, see GRP-105.

• 2008/07/07: v1.4 HEAD: Merge your grouper.properties with the grouper.example.properties. Pick up the new property: grouper.setters.dont.cause.queries

If you set this to true, refactor your the following methods:

- group.setAttribute(), group.setExtension(), group.setDescription(),
- group.setDisplayExtension() to call group.store() afterwards (could be after multiple calls)
- stem.setDescription(), stem.setExtension(), stem.setDisplayExtension(), to call stem.store() afterwards (could be after multiple calls)
- member.setSubjectSourceId(), member.setSubjectId(), to call member.store() afterwards (could be after multiple calls)

e.g. if the code is:

someGroup.setAttribute("whatever", "something");

change to:

someGroup.setAttribute("whatever", "something");
someGroup.store();

- 2008/06/30: v1.4 HEAD: Merge your grouper.properties with the grouper.example.properties. Pick up the new hooks configs, but leave the grouper.setters.dont.cause.queries setting out for now (or set to false).
- 2008/06/05: v1.4 HEAD: Remove the jar: i2mi-common-0.1.0.jar
- 2008/06/05: v1.4 HEAD: Add the jars (note these dont have version numbers since they were extracted from i2mi-common and Im not 100% sure of the version number... replacements will though):
  - commons-beanutils.jar
  - ° commons-collections.jar
  - ° commons-dbcp.jar
  - commons-digester.jar
  - commons-logging.jar
  - ° commons-pool.jar
  - dom4j.jar
  - ° odmg.jar

#### Grouper-ui

• 2008/09/23: v1.4 HEAD: <u>You can add tooltips to your group types and attributes</u> if you like. Add something like this to the custom nav.properties, though substitute the type name and attribute name for "grouperLoader" and "grouperLoaderdbName"

tooltipTargetted.groupTypes.grouperLoader=Group membership automatically managed via an external source, e.g. SQL query

tooltipTargetted.groupFields.grouperLoaderDbName=For sql based loader, this is the name of the db connection.

#### **Grouper-ws**

- 2008/12/4: v1.4.0rc1 HEAD: added WsSubject.identifierLookup. Note, for REST less than 1.4, this field will not be sent. Also, in wsAssignGrouperPrivilegesLiteResult, there was a privilegeResults field, which shouldne be there, it was removed. In wsAssignGrouperPrivilegesLiteResult the privilege type was all caps, it was changed to be all lower
- 2008/12/4: v1.4.0rc1 HEAD: Changed result codes for groupSave and stemSave. Instead of SUCCESS, it will be either: SUCCESS\_INSERTED, SUCCESS\_UPDATED, SUCCESS\_NO\_CHANGES\_NEEDED. Note, this is only for client version of 1.4+
- 2008/12/2: v1.4.0rc1 HEAD: Added two properties to grouper-ws.properties, compare with the example file
- 2008/11/6: v1.4 HEAD: Added two more fields to the wsGroupDetail object, adjust your clients appropriately:
  - compositeType (should be UNION, COMPLEMENT, INTERSECTION, or blank for non-composite)
     params
- 2008/10/28: v1.4 HEAD: some changes in conjunction with adding the privilege services
  - merge grouper-ws build.properties with example, and note the group.lib.dir has changed
     there was a weird situation where the result code was transmitted as an object...

<WsDeleteMemberLiteResult>

<resultMetadata>

<wsResultCode

#### class="edu.internet2.middleware.grouper.ws.soap.WsDeleteMemberLiteResult \$WsDeleteMemberLiteResultCode">SUCCESS</wsResultCode>

<resultCode>SUCCESS</resultCode> <resultMessage></resultMessage> <success>T</success>

I fixed this so it is not transmitted... clients should not have been using that element, but if so, check them. I believe this was only REST formats...

- 2008/10/21: v1.4 HEAD: <u>Axis2 1.3 was upgraded to Axis2 1.4</u>,
  - $^{\circ}\,$  you need to get all the lib/axis jars, and delete old dupes
    - replace or merge the axis.xml

° replace or merge the /services and /modules dirs

 2008/10/12: v1.4 HEAD: The grouper lib dir changed, so did this property in the grouperws.properties (note its not lib/grouper anymore): #e.g. from grouper-ws: lib

#### GSH

#### **Grouper-loader**

- 2008/12/12: v1.4.0: Merge grouper-loader.properties with grouper-loader.example.properties, there is a new setting for daily report directory
- 2008/12/9: v1.4.0 RC2 HEAD: if you are using grouper loader, and you are not auto creating types and attributes in grouper config, then add this new attribute with GSH:
  - subj=SubjectFinder.findById("GrouperSystem") sess=GrouperSession.start(subj) type=GroupType.createType(sess, "grouperLoader") read=Privilege.getInstance("read") admin=Privilege.getInstance("admin") type.addAttribute(sess, "grouperLoaderGroupQuery", read, admin, false)

• fd

### v1.4.0 Getting Started

This page last changed on Jan 04, 2009 by tbarton@uchicago.edu.

#### **<u>GROUPER</u>:** <u>About</u> <u>FAQ</u> <u>Software</u> <u>Documentation</u> <u>Archives</u> <u>Contribute</u> <u>WG</u> <u>Contact</u>

## Getting started with the Grouper API Binary Distribution v1.4.0

This page contains instructions for installing and configuring a new Grouper v1.4.0 installation.

- Meet required prerequisites and download the Grouper API binary release
- <u>API Building & Configuration</u>. Config files are in the conf/ directory. Config files ending in ".example" are templates and should not be modified. Config files not ending in ".example" are customizable
- Initialize your database (by default HSQLDB) by running gsh -registry -check -runscript

```
C:\dev_inst\eclipse\workspace_v33\grouper\bin>gsh -registry -check -runscript
Using GROUPER_HOME: C:\dev_inst\eclipse\workspace_v33\grouper\bin\..
Using GROUPER_CONF: C:\dev_inst\eclipse\workspace_v33\grouper\bin\../conf
Using JAVA:
                    "c:\dev_inst\java/bin/java"
                    64m-512m
using MEMORY:
Grouper starting up: version: 1.4.0 build date: 2008/11/13 14:42:25
grouper.properties read from: C:\dev_inst\eclipse\workspace_v33\grouper\conf\grouper.properties
Grouper current directory is: C:\dev_inst\eclipse\workspace_v33\grouper\bin
log4j.properties read from: C:\dev_inst\eclipse\workspace_v33\grouper\conf\log4j.properties
Grouper is logging to file: C:\dev_inst\eclipse\workspace_v33\grouper\bin\..\logs
\grouper_error.log, at min level WARN
for package: edu.internet2.middleware.grouper, based on log4j.properties
grouper.hibernate.properties: C:\dev_inst\eclipse\workspace_v33\grouper\conf
\grouper.hibernate.properties
grouper.hibernate.properties: sa@jdbc:hsqldb:C:\dev_inst\eclipse\workspace_v33\grouper\bin\..\dist/
run/grouper;create=true
sources.xml read from:
                             C:\dev_inst\eclipse\workspace_v33\grouper\conf\sources.xml
sources.xml groupersource id: g:gsa
sources.xml jdbc source id: jdbc: GrouperJdbcConnectionProvider
Based on grouper.properties: ddlutils.schemaexport.installGrouperData=true
(note, might need to type in your response multiple times (Java stdin is flaky))
(note, you can whitelist or blacklist db urls and users in the grouper.properties)
Are you sure you want to schemaexport all tables (dropThenCreate=F,writeAndRunScript=T) in
db user 'sa', db url 'jdbc:hsqldb:C:\dev_inst\eclipse\workspace_v33\grouper\bin\..\dist/run/
grouper;create=true'? (y|n):
У
Continuing...
Grouper ddl object type 'Grouper' has dbVersion: 0 and java version: 12
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:
C:\dev_inst\eclipse\workspace_v33\grouper\bin\..\ddlScripts\grouperDdl_20081114_02_00_20_771.sql
Script was executed successfully
```

C:\dev\_inst\eclipse\workspace\_v33\grouper\bin>

- Verify the installation was successful by examining log files in the logs/ directory.
- When using HSQLDB, a second attempt is sometimes necessary: gsh -registry -check -runscript
- It is now possible to run <u>GrouperShell (gsh)</u>and <u>Initialize Administration of Privileges</u>

C:\dev\_inst\eclipse\workspace\_v33\grouper\bin>gsh Using GROUPER\_HOME: C:\dev\_inst\eclipse\workspace\_v33\grouper\bin\.. Using GROUPER\_CONF: C:\dev\_inst\eclipse\workspace\_v33\grouper\bin\../conf Using JAVA: "c:\dev\_inst\java/bin/java" using MEMORY: 64m-512m Grouper starting up: version: 1.4.0 build date: 2008/11/13 14:42:25 grouper.properties read from: C:\dev\_inst\eclipse\workspace\_v33\grouper\conf\grouper.properties Grouper current directory is: C:\dev\_inst\eclipse\workspace\_v33\grouper\bin log4j.properties read from: C:\dev\_inst\eclipse\workspace\_v33\grouper\conf\log4j.properties Grouper is logging to file: console, at min level WARN for package: edu.internet2.middleware.grouper, based on log4j.properties grouper.hibernate.properties: C:\dev\_inst\eclipse\workspace\_v33\grouper\conf \grouper.hibernate.properties grouper.hibernate.properties: sa@jdbc:hsqldb:C:\dev\_inst\eclipse\workspace\_v33\grouper\bin\..\dist/ run/grouper;create=true sources.xml read from: C:\dev\_inst\eclipse\workspace\_v33\grouper\conf\sources.xml sources.xml groupersource id: g:gsa sources.xml jdbc source id: jdbc: GrouperJdbcConnectionProvider 2008-11-14 02:12:06,041: [main] WARN ApiConfig.printConfigOnce(189) - Grouper starting up: version: 1.4.0 build date: 2008/11/13 14:42:25 Type help() for instructions gsh 0% addRootStem("penn", "penn"); stem: name='penn' displayName='penn' uuid='b693f8f1-9763-44ae-a984-be25e0b6ea0f' gsh 1% addGroup("penn", "faculty", "faculty"); group: name='penn:faculty' displayName='penn:faculty' uuid='fbf3646f-eb10-47c1-9cc0-d32b7049a336' gsh 2%

GROUPER: About FAQ Software Documentation Archives Contribute WG Contact

## v1.4.1 Upgrade Instructions from v1.3.\*

This page last changed on Jan 29, 2009 by mchyzer.

## To upgrade from v1.4.0 grouper to v1.4.1, follow these instructions: (note the <u>changelog</u>)

#### API / Web Service / User Interface / Extensions

- Ldappc is now included with grouper, if you are going to use ldappc then get the latest binary or source release of grouper, setup the config files similar to how you did for 1.3.\*, and kickoff ldappc with gsh -ldappc
- If you are upgrading from 1.3.\* to 1.4.1, and you have already compared the grouper.properties file with grouper.example.properties, and run the ddl command, then you are done.
- If you upgraded to 1.4.0, and now you are upgrading to 1.4.1, continue:
- Get the latest grouper.jar, e.g. from the binary release
- Merge grouper.properties with grouper.example.properties, adding properties: junit.test.loader, junit.test.ldappc, junit.test.ddl
- Database conversion. The only change is the index on the name column of grouper\_stems is now unique. Here are 3 options for upgrade, pick one
  - Run: grouper\_home/bin/gsh -registry -check
    - Then look at the result script, and execute that script, perhaps with: grouper\_home/bin/gsh -registry -runsqlfile C:/mchyzer/isc/dev/grouper\_v1\_4/grouper/ ddlScripts/grouperDdl\_20090125\_08\_39\_44\_148.sql
  - -or- you can do this manually, change the stem name index to unique, and update grouper\_ddl table Grouper entry DB\_VERSION to 13:
  - update grouper\_ddl set db\_version = 13 where object\_name = 'Grouper'
  - -or- (not recommended) you could export the registry to backup, rebuild the registry with:
    - gsh -xmlexport GrouperSystem /whatever/20090125\_1\_4.xml
    - grouper\_home/bin/gsh -registry -drop -runscript
    - gsh -xmlimport GrouperSystem /whatever/20090125\_1\_4.xml

#### Web Services

• Get the latest grouper-ws.jar file

#### **Grouper Client**

• Merge the grouper.client.properties with grouper.example.properties. Note new entries for SSL, custom operations, and the new client version

## To upgrade from v1.3.\* grouper to v1.4, follow these instructions: (note the <u>changelog</u>)

#### API / Web Service / User Interface / Extensions

- Turn off access to your registry
- Backup your database and export your registry to XML as a backup: (e.g. ant xml-export -Dcmd="GrouperSystem export.xml")
- The libraries (jar file) have largely been renamed and upgraded. Delete all jars specific to grouper, and replace with the new list. You can get the latest jars from the lib subdirs of the binary release
- Merge all of your config files with the new example config files (you can obtain these from the conf directory of the binary release). While merging, read about the new settings, and pick your values.
  - Use a tool such as WinMerge (freeware), or Eclipse to make the task easy. Or, use the new example files, rename, and put your customizations in them
  - Merge grouper.properties with grouper.example.properties
     Note that grouper.properties was re-organized, so the order of the settings has changed
  - Merge grouper.hibernate.properties with grouper.hibernate.example.properties
  - Merge log4j.properties with log4j.example.properties
  - Merge sources.xml with sources.example.xml
  - Note you dont have to have JDBC credentials in this file if they are the same as grouper.hibernate.properties
  - Merge ehcache.xml with ehcache.example.xml

- Merge grouper.ehcache.properties with grouper.ehcache.example.properties
- Add grouper-loader.properties
- Add morphString.properties
- If you have a custom subject source, recompile it, there are new methods to implement
- In general recompile any code you have that uses Grouper/Subject references: e.g. package
- structure has changed, and MemberFinder no longer throws MemberNotFound exception...
  You should probably update your database driver jar... there are examples of recent ones in lib/ jdbcSamples... e.g. the error "composite\_composite\_idx in table grouper\_composites references the undefined column "owner"" is fixed by updating the postgres driver

#### Database conversion (try in test env first) TRACK 1 (do this or track 2)

- This is probably the easier way to go, though possibly more time consuming for large registries.
  - Also, you can try track 2, and if it isnt working for you, then switch to track 1.
    - Note, if you are using HSQL, you should get the newest driver available (e.g. from GROUPER\_HOME/lib/jdbcSamples). Upgrade your hsql probably. And do track 1 and not track 2.
- After exporting your data to xml (mentioned above), drop all the objects in your schema. Note, all your UUID's will change
- Run: qsh -registry -runscript
- This will create all the tables
- Then import your XML (e.g.):

gsh -xmlimport GrouperSystem /opt/grouper/1.3.1rc1/grouper/20081117-1.3.1.xml

#### Database conversion (try in test env first) TRACK 2 (do this or track 1)

Note, if you are using HSQL, you should get the newest driver available (e.g. from GROUPER\_HOME/lib/ jdbcSamples). Upgrade your hsql probably. And do track 1 and not track 2.

- This track will convert your data and structures to the new format
  - Look in your database (oracle or postgres only, not necessary for mysql) for unique constraints (not unique indices) on grouper tables (not necessary if Oracle). If any exist, remove them
     For oracle, execute the script generated by this query:

```
select 'alter table ' || table_name || ' drop constraint ' || constraint_name || ';' as the_command
from user_constraints
where constraint_type in ('R', 'U')
and (table_name like 'GROUPER_%' or table_name like 'SUBJECT%')
order by constraint_type, table_name;
```

 For postgres, execute the script generated by this query (note, substitute in your schema (public in case below), and catalog (grouper in the case below)

SELECT 'alter table ' || table\_name || ' drop constraint ' || constraint\_name || ';'
as the\_command
FROM information\_schema.table\_constraints where constraint\_type in ('FOREIGN
KEY', 'UNIQUE')
and lower(table\_name) like 'grouper%' and lower(table\_schema) = 'public' and
lower(table\_catalog) = 'grouper'

- order by constraint\_type, table\_name;
- For mysql, it shouldnt be necessary, but here is the script to generate the drop script (change table\_schema to match your schema, currently it is grouper):

```
SELECT concat(concat(concat('alter table ', table_name), ' drop foreign key '),
constraint_name), ';') as the_command
FROM information_schema.table_constraints where constraint_type in ('FOREIGN KEY')
and lower(table_name) like 'grouper%' and lower(table_schema) = 'grouper'
order by constraint_type, table_name;
```

- Run: gsh -registry -check (this command is in the bin dir of the binary release... make sure all your config files are configured)
  - Note: Seeing some errors here about tables missing, columns missing, ddl version mismatch is expected.

•

#### It will give output like this:

```
C:\dev_inst\eclipse\workspace_v33\grouper\bin>gsh -registry -check
Using GROUPER_HOME: C:\dev_inst\eclipse\workspace_v33\grouper\bin\..
Using GROUPER_CONF: C:\dev_inst\eclipse\workspace_v33\grouper\bin\../conf
Using JAVA:
                     "c:\dev_inst\java/bin/java"
using MEMORY:
                    64m-512m
Grouper starting up: version: 1.4.0 build date: 2008/11/13 14:42:25
grouper.properties read from: C:\dev_inst\eclipse\workspace_v33\grouper\conf\grouper.properties
Grouper current directory is: C:\dev_inst\eclipse\workspace_v33\grouper\bin
log4j.properties read from: C:\dev_inst\eclipse\workspace_v33\grouper\conf\log4j.properties
Grouper is logging to file: C:\dev_inst\eclipse\workspace_v33\grouper\bin\..\logs
\grouper_error.log, at min level WARN
for package: edu.internet2.middleware.grouper, based on log4j.properties
grouper.hibernate.properties: C:\dev_inst\eclipse\workspace_v33\grouper\conf
\grouper.hibernate.properties
grouper.hibernate.properties: grouper@jdbc:mysql://localhost:3306/grouper
sources.xml read from: C:\dev_inst\eclipse\workspace_v33\grouper\conf\sources.xml
sources.xml groupersource id: g:gsa
sources.xml jdbc source id: jdbc: GrouperJdbcConnectionProvider
Based on grouper.properties: ddlutils.schemaexport.installGrouperData=true
(note, might need to type in your response multiple times (Java stdin is flaky))
(note, you can whitelist or blacklist db urls and users in the grouper.properties)
Are you sure you want to schemaexport all tables (dropThenCreate=F,writeAndRunScript=F) in db user
 'grouper', db url 'jd
bc:mysql://localhost:3306/grouper'? (y|n):
Continuing...
Grouper ddl object type 'Grouper' has dbVersion: 0 and java version: 12
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:
C:\dev_inst\eclipse\workspace_v33\grouper\bin\..\ddlScripts\grouperDdl_20081114_01_11_45_007.sql
Note: this script was not executed per the grouper.properties:
ddlutils.schemaexport.writeAndRunScript
To run script via gsh, carefully review it, then run this:
gsh -registry -runsqlfile C:\\dev_inst\\eclipse\\workspace_v33\\grouper\\ddlScripts\
\grouperDdl_20081114_01_11_45_007.sql
```

C:\dev\_inst\eclipse\workspace\_v33\grouper\bin>

- Database conversion (continued)
  - Carefully examine the script (or have a SQL expert do this if you are not one) in that file (in this case: grouperDdl\_20081114\_01\_11\_45\_007.sql
  - Make sure all operations are ok. e.g. it should not drop any tables, or truncate tables, or anything else that looks bad.
  - Run this script in your DB IDE or grouper can do it with: gsh -registry -runsqlfile /wherever/ filename.sql
  - Note: If it has trouble dropping constraints or indexes, you can manually remove those. All the proper indexes (unless you had custom ones) will be recreated. Then continue the script where it left off.
- Turn on grouper, it should work. Look in the logs, make sure no configuration errors or warnings are thrown. If they are, address them.
- If you are using a source release of grouper, you should get the new source release (including build.xml, build.properties etc), and copy your properties files into the /conf dir. Merge the new build.example.properties with your version
- The database conversion script removed some cols, and relinked some foreign keys. The old data is still in the table. After you are satisfied grouper works correctly, you can remove these cols by changing the grouper.properties keys:

# after you have converted id's, and are happy with the conversion of removing the uuid col, # this will remove the backup uuid cols when running the gsh command: registryInitializeSchema() ddlutils.dropBackupUuidCols = true

# after you have converted field id foreign keys, and are happy with the conversion of removing the attribute name,

- # membership list name, and type cols,
- # this will remove the backup field name/type cols when running the gsh command:

registryInitializeSchema()
ddlutils.dropBackupFieldNameTypeCols = true

- Then generate the DDL again (the -deep option will examine the DB structure, not the DB version from the grouper\_ddl table): gsh -registry -deep
  - Then examine the resultant script carefully. Make sure the important operations (e.g. dropping/adding views are innocuous) are dropping the backup columns from the table
  - Turn off access to grouper
  - Backup your registry by database or xml export
  - Run that script
  - Verify the columns are removed
  - Grouper should work

#### Web services

- Axis jars have changed. Remove the old ones, add new ones from the grouper-ws release in the lib/ axis dir
- Nothing important has changed with grouper-ws.properties, but you may want to merge yours with grouper-ws.example.properties to be sure
- Replace or merge the axis.xml with the one from grouper-ws distribution
- Replace or merge the /services and /modules dirs from the grouper-ws distribution
- Two data changes were made to existing services:
  - Added two more fields to the wsGroupDetail object, adjust your clients appropriately (should only matter for SOAP clients or REST which are not coded correctly)
    - compositeType (should be UNION, COMPLEMENT, INTERSECTION, or blank for noncomposite)
    - params
  - There was a weird situation where the result code was transmitted as an object...this is not transmitted anymore

<WsDeleteMemberLiteResult>

- <resultMetadata>
- <wsResultCode

```
class="edu.internet2.middleware.grouper.ws.soap.WsDeleteMemberLiteResult
```

\$WsDeleteMemberLiteResultCode">SUCCESS</wsResultCode>

<resultCode>SUCCESS</resultCode>

<resultMessage></resultMessage>

```
<success>T</success>
```

• Merge grouper-ws build.properties with example, and note the group.lib.dir has changed

#### **User Interface**

 Merge the grouper nav.properties with yours. Note new properties about tooltips in types and attributes.

## v1.4 Release Notes

This page last changed on Feb 02, 2009 by tbarton@uchicago.edu.

GROUPER: About FAQ Software Documentation Archives Contribute WG Contact

## **Release Notes for Grouper v1.4.1**

Grouper v1.4.1 now includes Ldappc, and also has some minor fixes and improvements. See the <u>full list</u> <u>here</u>.

Release	Description of Feature Adds, Improvements, Changes, Revisions, & Fixes	Date
Grouper v1.4.1	Release v1.4.1 includes         the following fixes and         improvements:         • Improved: Ldappc is included         in the Grouper API and is         launched via Grouper Shell         • Improved: WS: replacing         existing members and actAs         • Improved: Prevent duplicate	2-February-2009

## **Release Notes for Grouper v1.4.0**

Grouper v1.4.0 has substantial new integration, automation, and diagnostic capabilities, enabling deployers to meet more requirements with less implementation effort. This page highlights some of the new features and fixes. Download Grouper v1.4.0 on the main <u>Download</u> page. Further below are <u>instructions for upgrading</u> from version 1.3.X and the <u>log of changes</u> from v1.3.1 to v1.4.0.

Details on the improvements and outstanding issues for this version can be found in the Internet2 Jira issue tracker for the <u>API</u>, the <u>UI</u>, and the <u>WS</u>.

Note: With release v1.4.0, building Grouper from source requires the ant compiler v1.7.0 or later.

Deleges		Data
Release	Description of Feature	Date
	Adds, Improvements,	
	Changes, Revisions, & Fixes	
Grouper v1.4.0	Release v1.4.0 includes	4-January-2009
_	the following fixes and	
	improvements:	
	<ul> <li>New: Programmatic hooks for local extensions to Grouper.</li> <li>New: Automatically maintain groups and memberships from external sources with <u>Grouper Loader</u>.</li> <li>Added: A <u>binary form</u> of the Grouper Toolkit.</li> <li>Improved: <u>Grouper Shell</u>, a comprehensive command line interface to Grouper and its allied utilities, is now a core part of the Grouper distribution. And there are correspondingly fewer ant targets in the source distribution.</li> </ul>	

## Upgrading from v1.3.X

# To upgrade from v1.4.0 grouper to v1.4.1, follow these instructions: (note the <u>changelog</u>)

#### API / Web Service / User Interface / Extensions

- Ldappc is now included with grouper, if you are going to use ldappc then get the latest binary or source release of grouper, setup the config files similar to how you did for 1.3.\*, and kickoff ldappc with gsh -ldappc
- If you are upgrading from 1.3.\* to 1.4.1, and you have already compared the grouper.properties file with grouper.example.properties, and run the ddl command, then you are done.
- If you upgraded to 1.4.0, and now you are upgrading to 1.4.1, continue:
- Get the latest grouper.jar, e.g. from the binary release
- Merge grouper.properties with grouper.example.properties, adding properties: junit.test.loader, junit.test.ldappc, junit.test.ddl
- Database conversion. The only change is the index on the name column of grouper\_stems is now unique. Here are 3 options for upgrade, pick one
  - Run: grouper\_home/bin/gsh -registry -check
    - Then look at the result script, and execute that script, perhaps with:

grouper home/bin/qsh -registry -runsqlfile C:/mchyzer/isc/dev/grouper v1 4/grouper/ ddlScripts/grouperDdl\_20090125\_08\_39\_44\_148.sql

- ° -or- you can do this manually, change the stem name index to unique, and update grouper\_ddl table Grouper entry DB\_VERSION to 13:
- update grouper ddl set db version = 13 where object name = 'Grouper'
- -or- (not recommended) you could export the registry to backup, rebuild the registry with:
  - qsh -xmlexport GrouperSystem /whatever/20090125\_1\_4.xml
  - grouper home/bin/gsh -registry -drop -runscript
  - gsh -xmlimport GrouperSystem /whatever/20090125\_1 4.xml

#### Web Services

· Get the latest grouper-ws.jar file

#### **Grouper Client**

• Merge the grouper.client.properties with grouper.example.properties. Note new entries for SSL, custom operations, and the new client version

## To upgrade from v1.3.\* grouper to v1.4, follow these instructions: (note the changelog)

#### API / Web Service / User Interface / Extensions

- Turn off access to your registry
- Backup your database and export your registry to XML as a backup: (e.g. ant xml-export -Dcmd="GrouperSystem export.xml")
- The libraries (jar file) have largely been renamed and upgraded. Delete all jars specific to grouper, and replace with the new list. You can get the latest jars from the lib subdirs of the binary release
- Merge all of your config files with the new example config files (you can obtain these from the conf directory of the binary release). While merging, read about the new settings, and pick your values.
  - Use a tool such as WinMerge (freeware), or Eclipse to make the task easy. Or, use the new example files, rename, and put your customizations in them
  - Merge grouper.properties with grouper.example.properties
  - Note that grouper properties was re-organized, so the order of the settings has changed Merge grouper.hibernate.properties with grouper.hibernate.example.properties

  - Merge log4j.properties with log4j.example.properties Merge sources.xml with sources.example.xml
  - Note you dont have to have JDBC credentials in this file if they are the same as grouper.hibernate.properties
  - Merge ehcache.xml with ehcache.example.xml
  - Merge grouper.ehcache.properties with grouper.ehcache.example.properties
  - Add grouper-loader.properties
  - Add morphString.properties
- If you have a custom subject source, recompile it, there are new methods to implement
- In general recompile any code you have that uses Grouper/Subject references: e.g. package structure has changed, and MemberFinder no longer throws MemberNotFound exception...
- You should probably update your database driver jar... there are examples of recent ones in lib/ jdbcSamples... e.g. the error "composite\_composite\_idx in table grouper\_composites references the undefined column "owner"" is fixed by updating the postgres driver

#### Database conversion (try in test env first) TRACK 1 (do this or track 2)

- This is probably the easier way to go, though possibly more time consuming for large registries.
  - Also, you can try track 2, and if it isnt working for you, then switch to track 1.
    - Note, if you are using HSQL, you should get the newest driver available (e.g. from GROUPER\_HOME/lib/jdbcSamples). Upgrade your hsql probably. And do track 1 and not track 2.
- After exporting your data to xml (mentioned above), drop all the objects in your schema. Note, all your UUID's will change
- Run: gsh -registry -runscript
- This will create all the tables
- Then import your XML (e.g.):

gsh -xmlimport GrouperSystem /opt/grouper/1.3.1rcl/grouper/20081117-1.3.1.xml

#### Database conversion (try in test env first) TRACK 2 (do this or track 1)

Note, if you are using HSQL, you should get the newest driver available (e.g. from GROUPER\_HOME/lib/ jdbcSamples). Upgrade your hsql probably. And do track 1 and not track 2.

- This track will convert your data and structures to the new format
  - Look in your database (oracle or postgres only, not necessary for mysql) for unique constraints (not unique indices) on grouper tables (not necessary if Oracle). If any exist, remove them
     For oracle, execute the script generated by this query:

select 'alter table ' || table\_name || ' drop constraint ' || constraint\_name || ';' as the\_command from user\_constraints where constraint\_type in ('R', 'U') and (table\_name like 'GROUPER\_%' or table\_name like 'SUBJECT%') order by constraint\_type, table\_name;

 For postgres, execute the script generated by this query (note, substitute in your schema (public in case below), and catalog (grouper in the case below)

SELECT 'alter table ' || table\_name || ' drop constraint ' || constraint\_name || ';'
as the\_command
 FROM information\_schema.table\_constraints where constraint\_type in ('FOREIGN
 KEY', 'UNIQUE')
 and lower(table\_name) like 'grouper%' and lower(table\_schema) = 'public' and
 lower(table\_catalog) = 'grouper'
 order by constraint\_type, table\_name;
 For myscal it shouldnt be percessary but here is the script to generate the drop script

- For mysql, it shouldnt be necessary, but here is the script to generate the drop script (change table\_schema to match your schema, currently it is grouper):

```
SELECT concat(concat(concat('alter table ', table_name), ' drop foreign key '),
constraint_name), ';') as the_command
FROM information_schema.table_constraints where constraint_type in ('FOREIGN KEY')
and lower(table_name) like 'grouper%' and lower(table_schema) = 'grouper'
order by constraint_type, table_name;
```

- Run: gsh -registry -check (this command is in the bin dir of the binary release... make sure all your config files are configured)
  - Note: Seeing some errors here about tables missing, columns missing, ddl version mismatch is expected.
  - It will give output like this:

```
C:\dev_inst\eclipse\workspace_v33\grouper\bin>gsh -registry -check
Using GROUPER_HOME: C:\dev_inst\eclipse\workspace_v33\grouper\bin\..
Using GROUPER_CONF: C:\dev_inst\eclipse\workspace_v33\grouper\bin\../conf
                    "c:\dev_inst\java/bin/java"
Using JAVA:
using MEMORY:
                    64m-512m
Grouper starting up: version: 1.4.0 build date: 2008/11/13 14:42:25
grouper.properties read from: C:\dev_inst\eclipse\workspace_v33\grouper\conf\grouper.properties
Grouper current directory is: C:\dev_inst\eclipse\workspace_v33\grouper\bin
log4j.properties read from: C:\dev_inst\eclipse\workspace_v33\grouper\conf\log4j.properties
Grouper is logging to file: C:\dev_inst\eclipse\workspace_v33\grouper\bin\..\logs
\grouper_error.log, at min level WARN
for package: edu.internet2.middleware.grouper, based on log4j.properties
grouper.hibernate.properties: C:\dev_inst\eclipse\workspace_v33\grouper\conf
\grouper.hibernate.properties
grouper.hibernate.properties: grouper@jdbc:mysql://localhost:3306/grouper
sources.xml read from:
                       C:\dev_inst\eclipse\workspace_v33\grouper\conf\sources.xml
sources.xml groupersource id: g:gsa
sources.xml jdbc source id: jdbc: GrouperJdbcConnectionProvider
Based on grouper.properties: ddlutils.schemaexport.installGrouperData=true
(note, might need to type in your response multiple times (Java stdin is flaky))
```

Are you sure you want to schemaexport all tables (dropThenCreate=F,writeAndRunScript=F) in db user 'grouper', db url 'jd bc:mysql://localhost:3306/grouper'? (y|n): У Continuing... Grouper ddl object type 'Grouper' has dbVersion: 0 and java version: 12 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:  $\texttt{C:} dev_inst|eclipse|workspace_v33|grouper|bin|..|ddlScripts|grouperDdl_20081114_01_11_45_007.sql|workspace_v33|grouper|bin|..|ddlScripts|grouperDdl_20081114_01_11_45_007.sql|workspace_v33|grouper|bin|..|ddlScripts|grouperDdl_20081114_01_11_45_007.sql|workspace_v33|grouper|bin|..|ddlScripts|grouperDdl_20081114_01_11_45_007.sql|workspace_v33|grouper|bin|..|ddlScripts|grouperDdl_20081114_01_11_45_007.sql|workspace_v33|grouper|bin|..|ddlScripts|grouperDdl_20081114_01_11_45_007.sql|workspace_v33|grouper|bin|..|ddlScripts|grouperDdl_20081114_01_11_45_007.sql|workspace_v33|grouper|bin|..|ddlScripts|grouperDdl_20081114_01_11_45_007.sql|workspace_v33|grouper|bin|..|ddlScripts|grouperDdl_20081114_01_11_45_007.sql|workspace_v33|grouper|bin|..|ddlScripts|grouperDdl_20081114_01_11_45_007.sql|workspace_v33|grouper|bin|..|ddlScripts|grouperDdl_20081114_01_11_45_007.sql|workspace_v33|grouper|bin|..|ddlScripts|grouper|bin|..|ddlScripts|grouper|bin|..|ddlScripts|grouper|bin|..|ddlScripts|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|grouper|bin|..|groupe$ Note: this script was not executed per the grouper.properties: ddlutils.schemaexport.writeAndRunScript To run script via gsh, carefully review it, then run this: gsh -registry -runsqlfile C:\\dev\_inst\\eclipse\\workspace\_v33\\grouper\\ddlScripts\ \grouperDdl\_20081114\_01\_11\_45\_007.sql

C:\dev\_inst\eclipse\workspace\_v33\grouper\bin>

- Database conversion (continued)
  - Carefully examine the script (or have a SQL expert do this if you are not one) in that file (in this case: grouperDdl\_20081114\_01\_11\_45\_007.sql
  - Make sure all operations are ok. e.g. it should not drop any tables, or truncate tables, or anything else that looks bad.
  - Run this script in your DB IDE or grouper can do it with: gsh -registry -runsqlfile /wherever/ filename.sql
  - Note: If it has trouble dropping constraints or indexes, you can manually remove those. All the proper indexes (unless you had custom ones) will be recreated. Then continue the script where it left off.
- Turn on grouper, it should work. Look in the logs, make sure no configuration errors or warnings are thrown. If they are, address them.
- If you are using a source release of grouper, you should get the new source release (including build.xml, build.properties etc), and copy your properties files into the /conf dir. Merge the new build.example.properties with your version
- The database conversion script removed some cols, and relinked some foreign keys. The old data is still in the table. After you are satisfied grouper works correctly, you can remove these cols by changing the grouper.properties keys:

# after you have converted id's, and are happy with the conversion of removing the uuid col, # this will remove the backup uuid cols when running the gsh command: registryInitializeSchema() ddlutils.dropBackupUuidCols = true

# after you have converted field id foreign keys, and are happy with the conversion of removing the attribute name,

# membership list name, and type cols,

# this will remove the backup field name/type cols when running the gsh command:

registryInitializeSchema()

ddlutils.dropBackupFieldNameTypeCols = true

- Then generate the DDL again (the -deep option will examine the DB structure, not the DB version from the grouper\_ddl table): gsh -registry -deep
  - Then examine the resultant script carefully. Make sure the important operations (e.g.
  - dropping/adding views are innocuous) are dropping the backup columns from the table Turn off access to grouper
  - Backup your registry by database or xml export
  - Run that script
  - Verify the columns are removed
  - Grouper should work

#### Web services

- Axis jars have changed. Remove the old ones, add new ones from the grouper-ws release in the lib/ axis dir
- Nothing important has changed with grouper-ws.properties, but you may want to merge yours with grouper-ws.example.properties to be sure
- Replace or merge the axis.xml with the one from grouper-ws distribution
- Replace or merge the /services and /modules dirs from the grouper-ws distribution
- Two data changes were made to existing services:
  - Added two more fields to the wsGroupDetail object, adjust your clients appropriately (should only matter for SOAP clients or REST which are not coded correctly)
    - compositeType (should be UNION, COMPLEMENT, INTERSECTION, or blank for noncomposite)
    - params
  - There was a weird situation where the result code was transmitted as an object...this is not transmitted anymore

```
<WsDeleteMemberLiteResult>
<resultMetadata>
<wsResultCode
class="edu.internet2.middleware.grouper.ws.soap.WsDeleteMemberLiteResult
$WsDeleteMemberLiteResultCode">SUCCESS</wsResultCode>
<resultCode>SUCCESS</resultCode>
<resultMessage></resultMessage>
<success>T</success>
```

· Merge grouper-ws build.properties with example, and note the group.lib.dir has changed

#### **User Interface**

• Merge the grouper nav.properties with yours. Note new properties about tooltips in types and attributes.

## **Grouper Change Log**

This document lists instructions for people with existing groups installations on how to upgrade to newer versions of grouper (or grouper related products). If you notice something missing please let us know.

The instructions are in descending order based on date/release. You will find instructions below for Grouper, Grouper-ws, Grouper-ui, GSH, USDU, etc. It is assumed if you are running grouper-ui that you will perform both the grouper upgrade notes, and the grouper-ui upgrade notes. It is understood that you will get the new source/javadoc/etc files, this document addresses configurations, jars, etc.

#### Grouper

- 2009/1/25: v1.4.1 GROUPER\_1\_4\_BRANCH: the stem name index was changed to unique from non-unique. Note, if you have problems adding this index, you need to remove stems with duplicate names from the registry. Obviously this needs to be done with care, and you might want to backup before attempting, or contact grouper-dev for help
  - Run: grouper\_home/bin/gsh -registry -check
    - Then look at the result script, and execute that script, perhaps with:
      - grouper\_home/bin/gsh -registry -runsqlfile C:/mchyzer/isc/dev/grouper\_v1\_4/grouper/ ddlScripts/grouperDdl\_20090125\_08\_39\_44\_148.sql
  - or- you can do this manually, change the stem name index to unique, and update grouper\_ddl table Grouper entry DB\_VERSION to 13
  - -or- (not recommended) you could export the registry to backup, rebuild the registry with run:
    - gsh -xmlexport GrouperSystem /whatever/20090125\_1\_4.xml
    - grouper\_home/bin/gsh -registry -drop -runscript
    - gsh -xmlimport GrouperSystem /whatever/20090125\_1\_4.xml
- 2009/1/25: v1.4.1 GROUPER\_1\_4\_BRANCH: merge grouper.properties with
  - grouper.example.properties, adding properties: junit.test.loader, junit.test.ldappc, junit.test.ddl
- 2008/11/14: v1.4 HEAD: compare and merge all config files with examples, things were rearranged, organized, etc
- 2008/10/29: v1.4 HEAD: update morphString.jar, algorithm changed
- 2008/10/14: v1.4 HEAD: update jars for config checking:

- <sup>°</sup> delete lib/grouper/commons-discovery-0.4.jar
- delete lib/grouper/jamon-2.7.jar
- update invoker.jar
- add commons-discovery.jar
- add jamon.jar
- update morphString.jar
- ° update p6spy.jar
- update subject.jar : note, if you have a custom source, you need to implement some new methods and recompile
- 2008/09/30: v1.4 HEAD: all jars have their version number removed, might want to delete old, and get new. e.g. from quartz-1.6.0.jar to quartz.jar
- 2008/09/23: v1.4 HEAD: <u>numParameters is now optional in sources.xml</u>. You can remove any numParameters params in sources.xml: <pre
- 2008/09/15: v1.4 HEAD: Subject API converted to use C3P0 pooling instead of DBCP.
  - delete: lib/commons-dbcp.jar
  - delete: lib/commons-pool.jar
  - update: lib/commons-logging.jar
  - delete: lib/subject-0.3.0-rc1-cvs.jar
  - add: lib/subject.jar
  - compare the sources.xml with the sources.example.xml, and add the elements and comments about jdbcConnectionProvider. You can leave the value blank for C3P0, or fill it in. If you are using the same credentials for grouper and a subject source, remove the credentials from sources.xml and put edu.internet2.middleware.grouper.subj.GrouperJdbcConnectionProvider as the jdbcConnectionProvider
- 2008/09/14: v1.4 HEAD: <u>Passwords encrypted in external files</u>. Add the jar: morphString.jar. Add morphString.example.properties, and morphString.properties. In sources.xml, replace the sourceadapter classes
  - FROM: edu.internet2.middleware.subject.provider.JDBCSourceAdapter TO: edu.internet2.middleware.grouper.subj.GrouperJdbcSourceAdapter
  - FROM: edu.internet2.middleware.subject.provider.JNDISourceAdapter TO: edu.internet2.middleware.grouper.subj.GrouperJndiSourceAdapter
  - If you want to take advantage of external encrypted passwords (optional):

- In morphString.properties, set the encrypt.key entry to a random alphanumeric string, or a pathname of a file containing the alphanumeric string

- In sources.xml, and grouper.hibernate.properties, encrypt the passwords with: java -jar morphString.jar, and put results in a file, and put the file path where the passwords were in sources.xml or grouper.hibernate.properties

- 2008/08/14: v1.4 HEAD: <u>UUID/ID cols reduced</u>, and <u>Field foreign key is now field id</u>. Delete any check constraints you have in your database. E.g. on the grouper\_fields table. Start grouper to see in the logs a message that says to run a script, or do an: ant schemaexport. You should review the script before running it, as sometimes ddlutils does weird things. Also, in prod you should export your data before running. Once you are comfortable with the result after running the script and checking your grouper data, set grouper.properties ddlutils.dropBackupUuidCols to true, and ddlutils.dropBackupFieldNameTypeCols to true, then: ant schemaexport, to get the script to drop the backup uuid/id cols. I would run all db scripts in prod with a DB ide (e.g. toad) or manually so that you can address any issues (see link for example issues)
- 2008/07/27: v1.4 HEAD: DDL management changed.

- Copy the DDL section from the grouper.example.properties into your grouper.properties, configure them

- Add the jar: ant-1.7.1.jar
- run: ant schemaexport to sync everything up (or generate the script to sync everything up, depending on grouper.properties)
  - Remove any schema-export.sql files
- 2008/07/22: v1.4 HEAD: If you were already using grouper loader, rename the table grouploader\_log to grouper\_loader\_log. If you already had the grouper\_ddl table, delete the java\_version column
- 2008/07/21: v1.4 HEAD: Add these jars: quartz-1.6.0.jar, commons-cli-1.1.jar, bsh-2.0b4.jar. GSH, loader, and usdu are in grouper now, with start scripts in grouper\_home/bin. See this jira report
- 2008/07/21: v1.4 HEAD: The java package structure was refactored. If you have code that doesnt compile, organize your imports. See this jira report
- 2008/07/20: v1.4 HEAD: Compare the grouper.properties with the grouper.example.properties, and add the section regarding group attribute validation via regex
- 2008/07/09: v1.4 HEAD: Everyone needs to tweak your log4j.properties:

Change FROM (remove .ErrorLog and .DebugLog):

log4j.logger.edu.internet2.middleware.grouper.ErrorLog	= ERROR, grouper_error
#log4j.logger.edu.internet2.middleware.grouper.DebugLog	= INFO, grouper_debug

TO:

log4j.logger.edu.internet2.middleware.grouper	= ERROR, grouper_error
<pre>#log4j.logger.edu.internet2.middleware.grouper</pre>	= INFO, grouper_debug

and REMOVE these lines entirely:

## Grouper Test Logging

log4j.logger.edu.internet2.middleware.grouper.TestLog

= INFO, grouper\_stdout

(probably doesn't affect you, but) If you have code that used the classes ErrorLog, DebugLog, or TestLog, these are removed, see GRP-105.

- 2008/07/07: v1.4 HEAD: Merge your grouper.properties with the grouper.example.properties. Pick up the new property: grouper.setters.dont.cause.queries
  - If you set this to true, refactor your the following methods:
    - group.setAttribute(), group.setExtension(), group.setDescription(),
    - group.setDisplayExtension() to call group.store() afterwards (could be after multiple calls) stem.setDescription(), stem.setExtension(), stem.setDisplayExtension(), to call
    - stem.store() afterwards (could be after multiple calls)
    - member.setSubjectSourceId(), member.setSubjectId(), to call member.store() afterwards (could be after multiple calls)

e.g. if the code is:

someGroup.setAttribute("whatever", "something");

change to:

someGroup.setAttribute("whatever", "something");
someGroup.store();

- 2008/06/30: v1.4 HEAD: Merge your grouper.properties with the grouper.example.properties. Pick up the new hooks configs, but leave the grouper.setters.dont.cause.queries setting out for now (or set to false).
- 2008/06/05: v1.4 HEAD: Remove the jar: i2mi-common-0.1.0.jar
- 2008/06/05: v1.4 HEAD: Add the jars (note these dont have version numbers since they were extracted from i2mi-common and Im not 100% sure of the version number... replacements will though):
  - ° commons-beanutils.jar
  - commons-collections.jar
  - ° commons-dbcp.jar
  - ° commons-digester.jar
  - ° commons-logging.jar
  - ° commons-pool.jar
  - ° dom4j.jar
  - ° odmg.jar

#### Grouper-ui

• 2008/09/23: v1.4 HEAD: <u>You can add tooltips to your group types and attributes</u> if you like. Add something like this to the custom nav.properties, though substitute the type name and attribute name for "grouperLoader" and "grouperLoaderdbName"

tooltipTargetted.groupTypes.grouperLoader=Group membership automatically managed via an external source, e.g. SQL query

tooltipTargetted.groupFields.grouperLoaderDbName=For sql based loader, this is the name of the db connection.

#### Grouper-ws

- 2008/12/4: v1.4.0rc1 HEAD: added WsSubject.identifierLookup. Note, for REST less than 1.4, this field will not be sent. Also, in wsAssignGrouperPrivilegesLiteResult, there was a privilegeResults field, which shouldne be there, it was removed. In wsAssignGrouperPrivilegesLiteResult the privilege type was all caps, it was changed to be all lower
- 2008/12/4: v1.4.0rc1 HEAD: Changed result codes for groupSave and stemSave. Instead of SUCCESS, it will be either: SUCCESS\_INSERTED, SUCCESS\_UPDATED, SUCCESS NO CHANGES NEEDED. Note, this is only for client version of 1.4+
- 2008/12/2: v1.4.0rc1 HEAD: Added two properties to grouper-ws.properties, compare with the example file
- 2008/11/6: v1.4 HEAD: Added two more fields to the wsGroupDetail object, adjust your clients appropriately:
  - compositeType (should be UNION, COMPLEMENT, INTERSECTION, or blank for non-composite)
     params
- 2008/10/28: v1.4 HEAD: some changes in conjunction with adding the privilege services
  - merge grouper-ws build.properties with example, and note the group.lib.dir has changed
  - there was a weird situation where the result code was transmitted as an object...

<WsDeleteMemberLiteResult>

<resultMetadata>

<wsResultCode

## class="edu.internet2.middleware.grouper.ws.soap.WsDeleteMemberLiteResult \$WsDeleteMemberLiteResultCode">SUCCESS</wsResultCode>

<resultCode>SUCCESS</resultCode> <resultMessage></resultMessage> <success>T</success>

I fixed this so it is not transmitted... clients should not have been using that element, but if so, check them. I believe this was only REST formats...

- 2008/10/21: v1.4 HEAD: Axis2 1.3 was upgraded to Axis2 1.4,
  - you need to get all the lib/axis jars, and delete old dupes
  - replace or merge the axis.xml
  - replace or merge the /services and /modules dirs
- 2008/10/12: v1.4 HEAD: The grouper lib dir changed, so did this property in the grouperws.properties (note its not lib/grouper anymore): #e.g. from grouper-ws: lib

#### GSH

#### **Grouper-loader**

- 2008/12/12: v1.4.0: Merge grouper-loader.properties with grouper-loader.example.properties, there is a new setting for daily report directory
- 2008/12/9: v1.4.0 RC2 HEAD: if you are using grouper loader, and you are not auto creating types and attributes in grouper config, then add this new attribute with GSH:

subj=SubjectFinder.findById("GrouperSystem")
sess=GrouperSession.start(subj)
type=GroupType.createType(sess, "grouperLoader")
read=Privilege.getInstance("read")
admin=Privilege.getInstance("admin")
type.addAttribute(sess, "grouperLoaderGroupQuery", read, admin, false)

• fd

Questions or comments? <sup>1</sup> Contact us.

**<u>GROUPER</u>: About FAQ Software Documentation Archives Contribute WG Contact</u>**
## Specsheet

This page last changed on Jan 04, 2009 by tbarton@uchicago.edu.

**<u>GROUPER</u>: About FAQ Software Documentation Archives Contribute WG Contact</u>** 

# **Grouper Product Specs as of v1.4.0**

Component	Details
RDBMS	<ul> <li>Object persistence is provided by Hibernate v3.2.6, which in turn uses JDBC to connect with a back-end RDBMS.</li> <li>DDL is generated by Hibernate. Grouper is currently tested against Oracle 9i, MySQL ?? and PostgreSQL 8 but should work with most databases supported by Hibernate.</li> </ul>
Java Servlet Container	<ul> <li>Servlet API Version 2.3.</li> <li>Known to run properly in Apache Tomcat 5.5.</li> <li>Have not tested/verified other servlet containers.</li> </ul>
Authentication	Through servlet container via REMOTE_USER, or via user-installable filter.
lava	Contributions: <u>Yale CAS authentication</u> filter.
Compiler	Ant v1.7.0 or later. Ensure that Ant can process Tomcat tasks by verifying that tools.jar (found in \$JAVA_HOME/lib) and catalina-ant.jar (in \$TOMCAT_HOME/server/lib) are in the classpath.
Browser Requirements	<ul> <li>XHTML 1.0</li> <li>CSS 2.1</li> <li>cookies must be enabled</li> <li>javascript must be enabled</li> </ul>

# **Grouper Product Specs as of v1.3.0**

Component	Details
RDBMS Java Servlet Container	<ul> <li>Object persistence is provided by Hibernate v3.2.6, which in turn uses JDBC to connect with a back-end RDBMS.</li> <li>DDL is generated by Hibernate. Grouper is currently tested against HSQLDB 1.7.2, Oracle 9i and PostgreSQL 8 but should work with most databases supported by Hibernate.</li> <li>Servlet API Version 2.3.</li> <li>Known to run properly in Apache Tomcat 5.5.</li> <li>Have not tested/verified other servlet containers.</li> </ul>
Authentication	Contributions: Yale CAS authentication filter.
Java	JDK v5.0.6 or later.
Compiler	Ant v1.6.3 or later.
Browser Requirements	<ul> <li>XHTML 1.0</li> <li>CSS 2.1</li> <li>cookies must be enabled</li> <li>javascript must be enabled</li> </ul>

# **Grouper Product Spec Sheet through v1.2.1**

Component	Details
RDBMS	<ul> <li>Object persistence is provided by Hibernate v2.1.8, which in turn uses JDBC to connect with a back-end RDBMS.</li> <li>DDL is generated by Hibernate. Grouper is currently tested against HSQLDB 1.7.2, Oracle 9i and PostgreSQL 8 but should work with most databases supported by Hibernate.</li> </ul>
Java Servlet Container	<ul> <li>Servlet API Version 2.3.</li> <li>Known to run properly in Apache Tomcat 4.1 and 5.5.</li> <li>Have not tested/verified other servlet containers.</li> </ul>
Authentication	Through servlet container via REMOTE_USER, or via user-installable filter.
Java	JDK v5.0.6 or later.
Compiler	Ant v1.6 or later.
Browser Requirements	<ul> <li>XHTML 1.0</li> <li>CSS 2.1</li> <li>cookies must be enabled</li> <li>no javascript needed, except for debug mode used only by UI developers</li> </ul>

### Reference

Java: If needed, download and install the latest version of JDK 1.5.0+ (5.0) from <u>http://java.sun.com/</u> <u>j2se/1.5.0</u>.

**Questions or comments? Contact us.** 

**<u>GROUPER:</u>** <u>About</u> <u>FAQ</u> <u>Software</u> <u>Documentation</u> <u>Archives</u> <u>Contribute</u> <u>WG</u> <u>Contact</u>

## **UI Building and Configuration**

This page last changed on Dec 05, 2007 by jbibbee@internet2.edu.

**<u>GROUPER</u>: About FAQ Software Documentation Archives Contribute WG Contact</u>** 

## Configuring and Deploying the Grouper UI v1.2.1

In this section we describe how to configure, build, and deploy the Grouper UI.

### **UI** Configuration

For many purposes, UI customization needs can be met by altering declarations in the grouper-ui/ resources/grouper/media.properties file. Logos, use of subject attributes in various search and display contexts, sorting behavior, and much more is specified in this file. See <u>Media Properties</u> for the details.

The UI is designed to be deeply customizable while remaining "upgrade proof". Readers needing all of the gory details should consult <u>Customising the Grouper UI</u>.

## **Building & Deploying**

- 1. Copy grouper-ui/build.properties.template to grouper-ui/build.properties.
- 2. Review grouper-ui/build.properties.
  - If you want the build script to automatically install the UI in your Tomcat instance, uncomment and set the appropriate value for deploy.home. If you do not set this you will need to copy the UI to your Tomcat installation's webapps directory. You will probably want to define the default.webapp.folder to suit how you intend to develop or customise the UI. See the <u>Grouper UI Development Environment</u> for options.
  - Make sure you set the grouper folder property to the location of your Grouper installation.
- 3. **Copy grouper-ui/template-tomcat-context.xml to grouper-ui/tomcat-context.xml** (or the value of the property deploy.context.xml if you have changed this).
  - Tomcat specific configuration can be added in this file e.g., container managed data sources.
- 4. Change directory to grouper-ui and type "ant".
  - A list of build targets is displayed. If you have set deploy.home enter "default". Otherwise type "dist" or "war". If the former copy <dist.home>/grouper to <TOMCAT\_HOME>/webapps, or if the latter, copy <dist.home>/grouper.war to <TOMCAT\_HOME>/webapps.
  - If you want to take advantage of the 'nice' targets you must uncomment and set appropriate values for all the deploy properties in grouper-ui/build.properties.

**Note:** The build process will attempt to create a directory peer to the grouper-ui directory. Hence, the directory grouper-ui/.. must be writable.

Questions or comments? <sup>1</sup> Contact us.

**<u>GROUPER</u>: About FAQ Software Documentation Archives Contribute WG Contact</u>** 

## **UI Customization Guide**

This page last changed on Dec 06, 2007 by jbibbee@internet2.edu.

## **GROUPER:** About FAQ Software Documentation Archives Contribute WG Contact

# Questions or comments? <sup>1</sup> Contact us.

Document	Description
Grouper UI Components	An overview of Grouper UI components.
Architecture	An explanation of the technologies used to create
	the Grouper UI and the specific approaches used.
Struts actions and tiles	An overview of the Struts actions and tiles defined
	by the Grouper UI.
Customising the Grouper UI	The QuickStart distribution contains UI
	customisations. This document explains those
	customisations - which can be used as the basis for
	your own site-specific customisations.
	Customisations may be limited to branding, page
	layout and text display, but can also include
	integrating authentication schemes and adding
	completely new functionality to integrate Grouper
	with other systems.
Grouper UI Development Environment	A description of the actual development environment
	used to develop the Grouper UI.
Grouper UI Contributed Code	A list of contributed code.

### **Unresolvable Subject Deletion Utility (USDU)**

This page last changed on Jan 04, 2009 by tbarton@uchicago.edu.

**GROUPER:** About FAQ Software Documentation Archives Contribute WG Contact

# Unresolvable Subject Deletion Utility (USDU)

This document is released alongside Grouper v1.4.0.

The Unresolvable Subject Deletion Utility finds and optionally deletes memberships for subjects which can not be found by their source.

An unresolvable subject is a subject that can not be found by its source. A subject may be unresolvable because of a temporary or permanent source failure, or because it was removed from its source before memberships or privileges were deleted or revoked.

This utility attempts to lookup every member's subject. If a subject can not be found, it's immediate memberships are printed and optionally deleted.

A future version may extend the Source class to provide more efficient lookups of subjects.

### Usage

\$GROUPER\_HOME/bin/gsh.sh -usdu <command line arguments>

Without any arguments, usdu prints its usage

usage: USDU -all	-source <arg>   -uuid <arg> [-delete] [-start <arg>]</arg></arg></arg>
-all	find unresolvable subjects from all sources
-delete	delete memberships and privileges
-source <arg></arg>	find unresolvable subjects from source
-start <arg></arg>	start session as this subject, default GrouperSystem
-uuid <arg></arg>	find unresolvable subject with member uuid

Unresolvable subjects are printed to stdout.

```
If an unresolvable subject is not a member of any groups:
    member_uuid='<uuid>' subject='<id>' no_memberships
```

For every group or stem and list that an unresolvable subject is a member of: member\_uuid='<uuid>' subject='<id>' group|stem='<name>' list='<name>' [delete]

For every unresolvable subject, usdu prints one line for every immediate memberbership. If an unresolvable subject is not a member of any groups and has no privileges, usdu prints no\_memberships.

#### Find unresolvable subjects from all sources

```
$GROUPER_HOME/bin/gsh.sh -usdu -all
member_uuid='e58697e4-...' subject='idl'/'source'/'person' group='stem:group1' list='members'
member_uuid='e58697e4-...' subject='id2'/'source'/'person' group='stem:group2' list='members'
...
```

#### Find unresolvable subjects from a specified source

\$GROUPER\_HOME/bin/gsh.sh -usdu -source CustomSource

#### Find unresolvable subject via member uuid

\$GROUPER\_HOME/bin/gsh.sh -usdu -uuid e58697e4-11a5-4082-b318-cb1e79191923

#### Delete unresolvable subject from all groups

```
$GROUPER_HOME/bin/gsh.sh -usdu -uuid e58697e4-... -delete
member_uuid='e58697e4-...' subject='idl'/'source'/'person' group='stem:group1' list='members'
delete
```

### Details

This utility finds and deletes memberships and privileges. It is possible for an unresolvable subject to be a creator or modifier of a group, in that case, calling Group.getCreateSubject() or Group.getModifySubject() will result in a SubjectNotFoundException.

Unresolvable subjects are not deleted from the grouper\_members table. If an unresolvable subject becomes resolvable again, it will retain its member uuid.

Questions or comments? <sup>1</sup> Contact us.

GROUPER: About FAQ Software Documentation Archives Contribute WG Contact