Import-Export legacy edition

XML Import / Export for Grouper v1.5 and before

Not all new features in 1.5 and later versions are exported or imported with this legacy import-export tool.

There is a new Import-Export utility in Grouper v1.6

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	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|^xml-tool-export.xsd|\.

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|^xml-tool-export-examples.html|\.

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:

Command	Summary of args.
args: -h	Prints this message
args:	subjectIdentifier [(-id -name -list)] [-ignoreInternal] [-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 ROOT 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|^xml-tool-import.xsd|\.

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. quickstart.xml|^xml-tool-quickstart.xml|\ 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

```
<list field="members"> <subject id="someId" source='someSource' type='person' immediate='true' /> </list>
```
