v1.0_Import-Export

This is an archived page. For the current version, please see Grouper XML Import/Export. Thanks!

Grouper XML Import / Export for Grouper v1.0

For version 1.0, Grouper includes XML import / export tools. Exported XML may be used for:

* provisioning to other systems
* reporting
* backups
* switching database backends - including to upgraded schemas (required by new Grouper API versions) in the same database

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.*
* initialising 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.

**Although data can be exported from one Grouper instance and imported into another, system attributes are not maintained. A group with the same name will have a different uuid and create times, etc, will reflect the time of import rather than the creation time in the original Grouper instance. A future version of the import tool may have options to maintain system attributes.

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 as an ant task in grouper/build.xml:

```
ant xml-export \-Dcmd="command line arguments"
```

The command line usage is:

Command	Summary of args.
args: -h	Prints this message

| args: | subjectIdentifier [(-id] [-name)] [-relative] [-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.
-name	The name of a Group or Stem to export.
-relative	If id or name specified do not export parent Stems.
-includeParent	If id or name identifies a Stem export this stem and child Stems or Groups.
filename	The file where exported data will be written. Will overwrite existing files.
properties	The name of a standard Java properties file which configures the export. Check Javadoc for a list of properties. If 'properties' is not specified, XmlExporter will look for 'export.properties' in the working directory. If this file does not exist XmlExporter will look on the classpath. If 'properties' is not specified and 'export.properties' cannot be found, the export will fail.

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 privileges 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 as an ant task in grouper/build.xml:

```
ant xml-import \-Dcmd="command line arguments"
```

The command line usage is:

Command	Summary of args.
args: -h	Prints this message
args:	subjectIdentifier [(-id -name -list)] 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. If no -id is specified, use=ROOT stem.
-name	The name of a Stem, into which, data will be imported. If no -name is specified, use=ROOT stem.
-list	File contains a flat list of Stems or Groups which may be updated. Missing Stems and Groups are not created.

filename	the file to import
properties	The name of a standard Java properties file which configures the import. Check Javadoc for a list of properties. If 'properties' is not specified, XmlImporter will look for 'import.properties' in the working directory. If this file does not exist XmlImporter will look on the classpath. If 'properties' is not specified and 'import.properties' cannot be found and import options are not included in the XML, the import will fail.

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. quickstart.xml is a minimal XML import file which creates the demo registry*.

*For Grouper 0.9 and earlier, the demo registry was created by loading an XML file using a contributed XmlLoader class. The format of XML recognised by XmlLoader is significantly different to the *official* format understood by XmlImporter, however, if created files in the old format, these can still be loaded at the Root stem, simply add <data></data> inside the <registry> tags but outside your actual Styem and Group data. You can then export the registry to obtain an XML file in the new format. The XmlLoader format will not be supported in the next Grouper release.

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.

Questions or comments? Contact us.