

Managing Grouper in several environments

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

The Grouper distribution has a build script for one environment. If you need to manage dev/test/prod for the loader/ui/ws that is 9 environments. When you need to upgrade, it can be time consuming and easy to make mistakes.

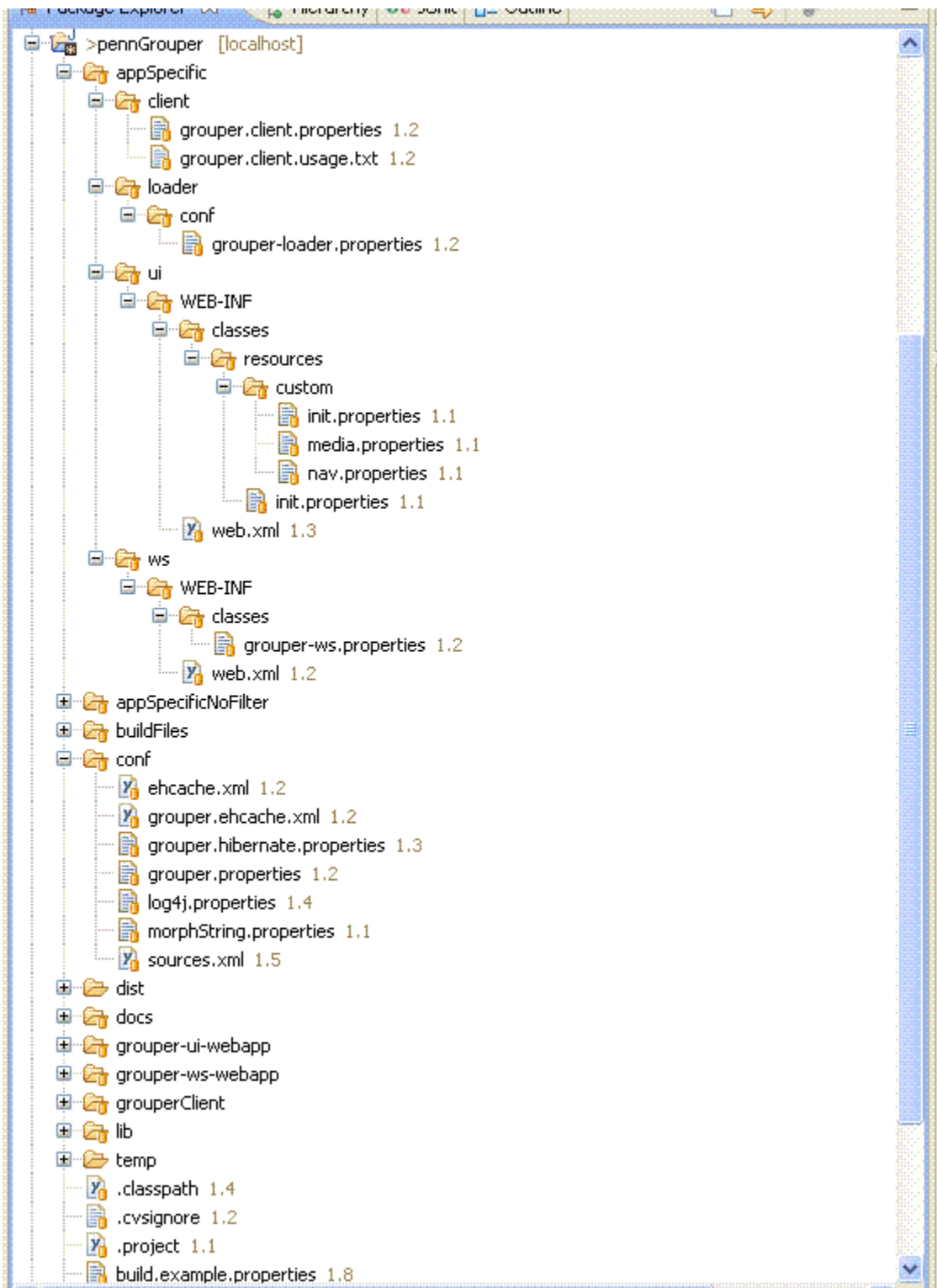
This is a strategy we use at Penn to manage all the config files and build scripts for all the environments, without duplication, and storing files in CVS. This is probably not a strategy that can be used without modification, but you can use the design and tweak the scripts for your environments.

Note that this uses ant, not Maven.

Here is the example output. Note the UI and WS zips just have a warfile inside, but since the warfiles in this case (configurable in build.properties) have the same name, so we zip them in different names so they can live in the same dir.

- dist/grouperUiDev.tar.gz
- dist/grouperUiTest.tar.gz
- dist/grouperUiProd.tar.gz
- dist/grouperWsDev.tar.gz
- dist/grouperWsTest.tar.gz
- dist/grouperWsProd.tar.gz
- pennGroupsClient-1.5.2.tar.gz

Here is the folder structure:



- **appSpecific**: holds files for grouper a grouper component that is different from a grouper build, and different in various envs. Generally config files or UI customizations for an institution
- **appSpecificNoFilter**: holds files like appSpecific, but will be copied without filtering (search/replace) variables (e.g. the passwords in each env differ). These are files without variables, or binary files
- **buildFiles**: holds build.properties for Grouper builds
- **conf**: common grouper API config files which are different than the default or which differ in various environments
- **dist**: the builds go there
- **docs**: documentation
- **grouper-ui-webapp**: a build of the UI without files which different for institution or files which different per env
- **grouper-ws-webapp**: a build of the WS without files which different for institution or files which different per env
- **grouperClient**: the binary grouper client without files specific for an institution

- lib: ant library for it's in build script
- temp: used for builds
- the [build.xml](#) and [build.example.properties](#) are in the base dir

Setup and Maintenance

This section describes a new setup, or an upgrade (the CVS syncs)

- Create that folder structure above e.g. in C:\dev\ eclipse\ projects\ pennGrouper
- Download the later Grouper API binary, UI, WS, and client (binary)
 - E.g. in C:\software\ grouper\ 1.5.2\ grouper.apiBinary-1.5.2
 - E.g. in C:\software\ grouper\ 1.5.2\ grouper.clientBinary-1.5.2
 - E.g. in C:\software\ grouper\ 1.5.2\ grouper.ui-1.5.2
 - E.g. in C:\software\ grouper\ 1.5.2\ grouper.ws-1.5.2
- Copy all the build.example.properties to build.properties in each of the 4. Look at the build.properties, copy the file to the buildFiles dir. Sync with CVS and diff to see if there are changes (if not new). If there are changes, then copy the file back to its original location
- Example of changes, in the WS and UI build.properties, put the path to Grouper in there.
- Run "ant dist" in the UI and WS
- Edit this build.properties, and put in the locations to everything

```
grouper.dir=C:\\software\\grouper\\1.5.2\\grouper.apiBinary-1.5.2

grouper.jar.dir=${grouper.dir}/dist/lib

grouper.client.dir=C:\\software\\grouper\\1.5.2\\grouper.clientBinary-1.5.2

grouper.ui.dir=C:\\software\\grouper\\1.5.2\\grouper.ui-1.5.2\\dist\\grouper

grouper.ws.dir=C:\\software\\grouper\\1.5.2\\grouper.ws-1.5.2\\grouper-ws\\build\\dist\\grouper-ws
```

- Run the ant task: copyGrouperApiFilesToBuildProject. This will copy the grouper config files to the conf directory. If this is an upgrade, compare the files with the latest in CVS and merge your changes. Otherwise look at the files and make changes as necessary. However, if you have something which differs per env, put a variable in there. In this case, here are my variables in grouper.properties:

```
grouper.env.name = <at:var at:name="envName" />
```

There are corresponding entries in the build.properties:

```
devEnv=DEV
testEnv=TEST
prodEnv=PROD
```

Here are the variables in the log4j.properties:

```
log4j.appender.grouper_error.File = <at:var at:name="logDir" />/grouper_error.log
```

This is the one place where the data is different per env and grouper module (well, mine are similar since I run on different servers, but they can be different). Here are the build.properties entries

```
devUiLogDir=/opt/appserv/tomcat_2v/logs/grouper
testUiLogDir=/opt/appserv/tomcat_2v/logs/grouper
prodUiLogDir=/opt/appserv/tomcat_2v/logs/grouper

devWsLogDir=/opt/appserv/tomcat_3b/logs/grouper/grouper_error.log
testWsLogDir=/opt/appserv/tomcat_3b/logs/grouper/grouper_error.log
prodWsLogDir=/opt/appserv/tomcat_3b/logs/grouper/grouper_error.log

localdevProjectLogDir=c:/temp
devProjectLogDir=/opt/appserv/tomcat_3c/logs/grouper
testProjectLogDir=/opt/appserv/tomcat_3c/logs/grouper
prodProjectLogDir=/opt/appserv/tomcat_3c/logs/grouper
```

Here are the grouper.hibernate.properties variables

```
hibernate.connection.url = <at:var at:name="dbUrl" />

hibernate.connection.username      = <at:var at:name="dbUser" />

hibernate.connection.password     = <at:var at:name="dbPass" />
```

There are entries per env. Note the passwords are encrypted with the morphString Internet2 library, so the encrypted values are in a file system file (better for storage of config files in CVS nad hiding plaintext passwords)

```
devDbUrl=jdbc:oracle:thin:@devserver:1521:devsid
testDbUrl=jdbc:oracle:thin:@testserver:1521:testsid
prodDbUrl=jdbc:oracle:thin:@prodserver:1521:prodsid

devDbUser=myuser
testDbUser=myuser
prodDbUser=myuser

localdevDbPass=r:/home/appadmin/pass/grouper/grouperMorphDev.pass
devDbPass=/home/appadmin/pass/grouper/grouperMorphDev.pass
testDbPass=/home/appadmin/pass/grouper/grouperMorphTest.pass
prodDbPass=/home/appadmin/pass/grouper/grouperMorphProd.pass
```

grouper-loader.properties (in appSpecific/conf instead of conf since only needed in the loader) also has a variable for the report directory and loader servers and passes:

```
db.warehouse.pass = <at:var at:name="warehousePass" />
db.warehouse.url = <at:var at:name="warehouseUrl" />
daily.report.saveInDirectory = <at:var at:name="dailyReportDirectory" />
```

These are in the build.properties

```
localdevWarehousePass=r:/home/appadmin/pass/grouper/grouperMorphWarehouse.pass
devWarehousePass=/home/appadmin/pass/grouper/grouperMorphWarehouse.pass
testWarehousePass=/home/appadmin/pass/grouper/grouperMorphWarehouse.pass
prodWarehousePass=/home/appadmin/pass/grouper/grouperMorphWarehouse.pass

localdevWarehouseUrl=jdbc:oracle:thin:@testwarehouse:1521:testwhsid
devWarehouseUrl=jdbc:oracle:thin:@testwarehouse:1521:testwhsid
testWarehouseUrl=jdbc:oracle:thin:@testwarehouse:1521:testwhsid
prodWarehouseUrl=jdbc:oracle:thin:@prodwarehouse:1521:prodwhside

localdevDailyReportDirectory=c:/temp/groupDailyReports
devDailyReportDirectory=/opt/appserv/tomcat_3c/logs/fastGrouper/dailyReports
testDailyReportDirectory=/opt/appserv/tomcat_3c/logs/fastGrouper/dailyReports
prodDailyReportDirectory=/opt/appserv/tomcat_3c/logs/fastGrouper/dailyReports
```

UI build scripts

Run the ant task: copyGrouperUiToBuildProject (note this assumes you ran the UI build target: dist). This will copy the build of the UI to the pennGrouper dir, but leaves out the grouper config files (since those are in the conf dir).

You can run the ant task: packageUi which will build the three UI warfiles and zips for dev, test, and prod

WS build scripts

Run the ant task: copyGrouperWsToBuildProject. This will copy the build of the WS to the pennGrouper dir, but leaves out the grouper config files (since those are in the conf dir). Note, if this is an upgrade, sync the appSpecific\ws\WEB-INF\classes\grouper-ws.properties and compare and merge with old values.

You can run the ant task: packageWs which will build the three WS warfiles and zips for dev, test, and prod

Client build scripts

See the setting in the build.properties to control the name of the client zip

```
clientPackageName=pennGroupsClient-1.5.2.tar.gz
```

Run this target: copyGrouperClientToBuildProject. This copies the binary client to the grouperClient dir so it can be saved in CVS. If this is an upgrade, compare the appSpecific\client\grouper.client.properties and appSpecific\client\grouper.client.usage.txt and put in the institution specific settings.

Run the target: packageClient to build the client zip for distribution in your institution. Note there isnt one for each env since the user can adjust the URL to connect to. Might want to give some examples of test envs in the properties file comments, unless you dont want to for security or to not confuse people.

Loader project build scripts

At Penn we run the loader in a webapp. So we need the grouper API and config files in the webapp. That webapp has dev/test/prod envs so we need to make sure the config files are correct. Configure the location of the project in the build.properties, the config dirs for each env, and also the log directories for the 4 envs

```
externalProjectLibDir=C:\\dev\\eclipse\\projects\\fastGrouper\\sourcecontrol\\web\\WEB-INF\\lib

localdevConfDir=C:\\dev\\eclipse\\projects\\fastGrouper\\resources
devConfDir=C:\\dev\\eclipse\\projects\\fastGrouper\\sourcecontrol\\servers\\integratedDev\\resources
testConfDir=C:\\dev\\eclipse\\projects\\fastGrouper\\sourcecontrol\\servers\\testDaemon\\resources
prodConfDir=C:\\dev\\eclipse\\projects\\fastGrouper\\sourcecontrol\\servers\\prodDaemon\\resources

localdevProjectLogDir=c:/temp
devProjectLogDir=/opt/appserv/tomcat_3c/logs/grouper
testProjectLogDir=/opt/appserv/tomcat_3c/logs/grouper
prodProjectLogDir=/opt/appserv/tomcat_3c/logs/grouper
```

Run the ant task: copyBuildProjectToWebapp which will copy the grouper jars to the lib dir, and the config files to each location above (and substitute variables). The building of the webapp is done by that application.