Grouper configuration files and overlays

In a patch to Grouper 2.4, and in Grouper 2.5+, Grouper allows configuration to be stored in the database rather than in configuration files. This is the recommended approach.
See details here.

Some Grouper configuration files can have overlays so that only the changes from the defaults of the config files need to be tracked in the institution specific config file. Also, configs can be centrally stored on a server across multiple webapps or standalone Grouper applications. There can be a default configuration file, and an override file so that only the changes from the default can be tracked in the overlay.

Using this approach to configuration files can make Grouper more easily deployable across environments, and more easily upgradable.

This is available in Grouper v2.2+ for the following config files:

- grouper.properties
- grouper.hibernate.properties
- grouper-loader.properties
- grouper-ui.properties
- grouper-ws.properties
- grouper.client.properties
- subject.properties
- grouper.cache.properties (2.3.0.patch+)
- grouper.text.en.us.properties

In the future we can add this feature to other config files as well.

Java code to read configs from config files (e.g. for GSH templates or scripts)

```
import java.util.Map;
import java.util.Set;
import java.util.regex.Pattern;

import edu.internet2.middleware.grouper.GrouperSession;
import edu.internet2.middleware.grouper.app.loader.GrouperLoaderConfig;
import edu.internet2.middleware.grouper.cfg.GrouperConfig;
import edu.internet2.middleware.grouper.cfg.GrouperHibernateConfig;
import edu.internet2.middleware.grouper.cfg.text.GrouperTextContainer;
import edu.internet2.middleware.grouper.ui.util.GrouperUiConfigInApi;
import edu.internet2.middleware.grouper.ws.GrouperWsConfigInApi;
import edu.internet2.middleware.grouperClient.config.GrouperUiApiTextConfig;

public class Test63main {

  public static void main(String[] args) {

    GrouperSession.startRootSession();

    boolean defaultBoolean = false;
    GrouperConfig.retrieveConfig().propertyValueBoolean("key", defaultBoolean);
    GrouperConfig.retrieveConfig().propertyValueBooleanRequired("key");
    GrouperConfig.retrieveConfig().propertyValueString("stringKey");
    GrouperConfig.retrieveConfig().propertyValueStringRequired("stringKey");
    GrouperConfig.retrieveConfig().propertyValueString("stringKey2", "defaultValue");
    int defaultInt = 999;
    GrouperConfig.retrieveConfig().propertyValueInt("intKey", defaultInt);
    GrouperConfig.retrieveConfig().propertyValueInt("intKey");
    GrouperConfig.retrieveConfig().propertyValueIntRequired("intKey");

    Map<String, String> propertiesMap = GrouperHibernateConfig.retrieveConfig().propertiesMap(Pattern.compile
("^something\\.([^.]+)\\..*$"));

    Set<String> propertyConfigIds = GrouperLoaderConfig.retrieveConfig().propertyConfigIds(Pattern.compile
("^something\\.([^.]+)\\..*$"));

    GrouperUiConfigInApi.retrieveConfig().propertyValueString("key");

    GrouperWsConfigInApi.retrieveConfig().propertyValueString("key");

    // this will eval all jexl scripts.  if you want the jexl script to show on screen use HTML for dollar: $
    GrouperTextContainer.textOrNull("someUiKey");

    // dont eval jexl scripts, just get the raw value
    GrouperUiApiTextConfig.retrieveTextConfig().propertyValueString("someKey");
  }

}
```

Escape chars

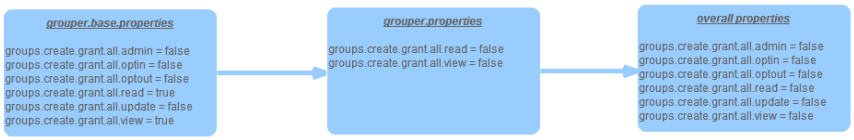
You can escape chars, when they are read by the property config framework, they will be unescaped

Unicode	Value
U+0024	$
U+0020	space
U+007B	{
U+007D	}

U+000A	newline \n
U+002B	+

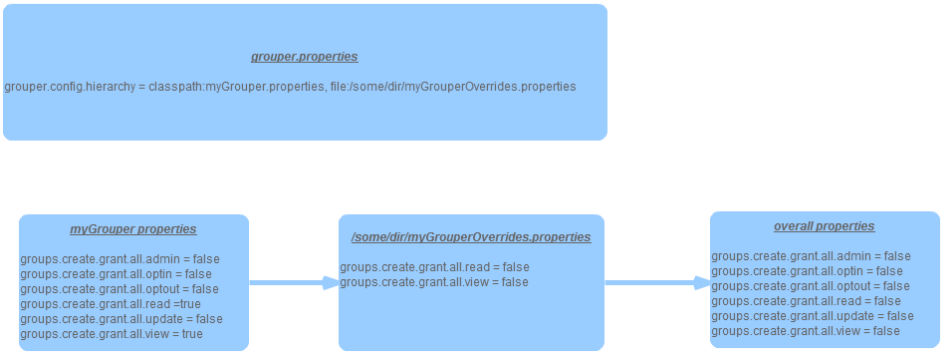
How it works

Each of the properties files has a base file, and a config file. For example, there is a grouper.base.properties, and a grouper.properties. Both of these are located on the classpath in the default package. e.g. WEB-INF/classes/grouper.base.properties and WEB-INF/classes/grouper.properties. Generally all the default settings will be located in the base file, and only the things that are overridden are in the grouper.properties. This is a change in Grouper v2.2+ since before that all properties are in the grouper.properties file, and the example file was used just to show what configs are possible.



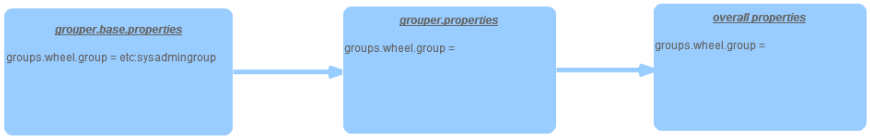
Specify the hierarchy

If you do not want to use the base and the overlay in the classpath, you can specify which files are used for the properties. This must be specified in the base or config file. Each config file has its own key for this hierarchy, but it is listed in the base config file. Here is an example for the grouper. properties. Note, it is recommended to include the classpath base properties, though it is up to you. You can specify config files by classpath or file location.



Edge cases

If you override a key with an empty value, that will blank out that config value



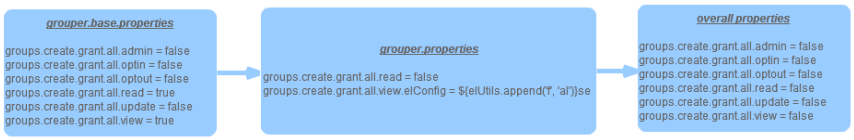
Config file reloading

You can specify the number of seconds that the config file will be checked to see if there are differences. This is not a trivial check, so it is recommended not to be more often than every 60 seconds. If -1 is specified, then it will never be checked. If it is 0 then it will be checked each time a config param is referenced. Note, this is another properties that must be specified in the base or config file, it cannot be put in other places specified in the hierarchy. Here is an example for grouper.properties:

```
grouper.config.secondsBetweenUpdateChecks = 60
```

Expression language in property values

You can specify the property key in such a way that you can have value include expression language scriplets. The class edu.internet2.middleware.grouperClient.util.GcElUtilsSafe can be referenced as "elUtils". To specify a key as EL, append this suffix to the config key: .elConfig



To use a custom class in the EL, Write a class with a static method, compile, put in a jar on the classpath. There should be a default constructor in the class. Refer to the fully qualified class in EL:

```
package edu.internet2.middleware.grouperClient.config;

/**
 * some test class for EL
 * @author mchyzer
 *
 */
public class SomeTestElClass {

  /**
   * some method for EL
   * @param a
   * @param b
   * @return the result
   */
  public static String someMethod(String a, String b) {
    return a + b + " something else";
  }

}
```

EL in a properties file:

```
some.config.2.elConfig = ${edu.internet2.middleware.grouperClient.config.SomeTestElClass.someMethod('start', '
middle')}
```

This will result in the value (for key: some.config.2): start middle something else

Another example, look in another non-grouper properties file

```
hibernate.connection.url.elConfig = ${ edu.internet2.middleware.grouper.util.GrouperUtil.propertiesFromFile(new
("java.io.File","/Users/mchyzer/git/grouper_v2_5/grouper/conf/myfile.properties"), false).getProperty("myurl") }
```

Environment variables

If you want a grouper properties config file to have an env var, you can have any property be an env var... just configure it in a config file as (for property a.b.c):

```
a.b.c.elConfig = ${elUtils.processEnvVarOrFile('SOME_ENV_VAR')}
```

That will cause the property "somethingWhatever" to have the value "c:\dev_inst\java" or whatever it is set to. Do the following two things:

1. append .elConfig to the propertyName
2. this is the value: ${elUtils.processEnvVarOrFile('JAVA_HOME')}
3. note, if the value of the env var is a variable, it will get it from there

Example:

When I have this in grouper.properties:

```
# in cases where grouper is logging or emailing, it will use this to differentiate test vs dev vs prod
grouper.env.name = GROUPERDEMO_2_2_2
```

I can print it out in gsh:

```
gsh 1% edu.internet2.middleware.grouper.cfg.GrouperConfig.retrieveConfig().propertyValueString("grouper.env.
name")
GROUPERDEMO_2_2_2
```

Add an env variable:

```
[appadmin@i2midev1 bin]$ export GROUPER_ENV=GROUPER_2_2_2_fromEnv
[appadmin@i2midev1 bin]$ echo $GROUPER_ENV
GROUPER_2_2_2_fromEnv
```

Change grouper.properties

```
# grouper.env.name = GROUPERDEMO_2_2_2
grouper.env.name.elConfig = ${java.lang.System.getenv().get('GROUPER_ENV')}
```

Restart GSH, try again

```
gsh 0% edu.internet2.middleware.grouper.cfg.GrouperConfig.retrieveConfig().propertyValueString("grouper.env.
name") GROUPER_2_2_2_fromEnv
```

Misc

You can use these special vars: $space$ and $newline$ to represent a space or newline.

Refer to other properties in that config file

In both of these cases the value for the property "somethingWhatever1" is: **prefix something to be reused suffix**

Pre 2.3.0 patched, you can do this:

```
somethingWhatever = something to be reused
somethingWhatever1.elConfig = prefix ${edu.internet2.middleware.grouperClient.util.GrouperClientConfig.
retrieveConfig().propertyValueString("somethingWhatever")} suffix
```

Post 2.3.0 you can do this:

```
somethingWhatever = something to be reused
somethingWhatever1 = prefix $$somethingWhatever$$ suffix
```

New Configuration File: subject.properties

Note: subject.properties is a new config file in Grouper 2.2 that specifies the location of the sources.xml file.

Use Case

Some sites may choose to have two levels of overlay files above the base file, as follows:

- grouper.base.properties (unmodified built in properties)

- grouper.properties (institution-wide)
- grouper.local.properties (specific to institution and environment, e.g. the “test” env overlays)

See Also

For details on migrating to using the Configuration Overlay files, see Upgrade Instructions to Grouper 2.2 from Grouper 2.1

API Building and Configuration

When I have this in grouper.properties:

in cases where grouper is logging or emailing, it will use this to differentiate test vs dev vs prod

grouper.env.name = GROUPERDEMO_2_2_2

I can print it out in gsh:

gsh 1% edu.internet2.middleware.grouper.cfg.GrouperConfig.retrieveConfig().propertyValueString ("grouper.env.name")

GROUPERDEMO_2_2_2

Add an env variable:

[appadmin@i2midev1 bin]$ export GROUPER_ENV=GROUPER_2_2_2_fromEnv

[appadmin@i2midev1 bin]$ echo $GROUPER_ENV

GROUPER_2_2_2_fromEnv

Change grouper.properties

grouper.env.name = GROUPERDEMO_2_2_2

grouper.env.name.elConfig = ${java.lang.System.getenv().get('GROUPER_ENV')}

Restart GSH, try again

gsh 0% edu.internet2.middleware.grouper.cfg.GrouperConfig.retrieveConfig().propertyValueString ("grouper.env.name")

GROUPER_2_2_2_fromEnv

See Also

Grouper Configuration in the Database and UI