

Grouper internationalization

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

This page presents information on Grouper and internationalization, handling non-English characters, UTF-8, accents, etc.

Grouper and non-English characters

Grouper works with non-English characters, but the environment must be setup correctly for this to happen. Your database must support it, your Java settings must be set correctly, your Tomcat settings must be set correctly. In Grouper 2.2.1 patch grouper_v2_2_1_api_patch_9 there are checks at startup. Grouper patches are listed on the [Release Notes page](#).

There are checks at startup to make sure the non-Tomcat settings are correct.

Also, the default for grouper.base.properties for file encoding was changed. If you don't need non-English chars and you are worried about the risk of this change, please change it in grouper.properties

```
# when reading writing files from util classes, this is encoding (was ISO-8859-1)
grouper.default.fileEncoding = UTF-8
```

These settings will detect if your database and file system can handle UTF-8, and if transactions, and case sensitivity are handled correctly. You can get a success message if you are debugging, but this defaults to off. These will run in another thread so they don't affect start time (optionally default to true)

```
# if should check database and utf in new thread
configuration.checkDatabaseAndUtf.inNewThread = true
# if grouper should check to see if the database has case sensitive selects
configuration.detect.db.caseSensitive.problems = true
configuration.display.db.caseSensitive.success.message = false
# if grouper should check to see if utf-8 works on startup in files
configuration.detect.utf8.file.problems = true
# if grouper should check to see if utf-8 works on startup in the database
configuration.detect.utf8.problems = true
configuration.display.utf8.success.message = false
# if grouper in the utf8 check will check to see if grouper supports transaction
configuration.detect.db.transaction.problems = true
configuration.display.transaction.success.message = false
```

If there is a problem with start, you will see a message like this, but it will not cause grouper not to start:

```
Error: Cannot properly read UTF string from database: '??????', make sure your database has UTF tables and
perhaps a hibernate.connection.url in grouper.hibernate.properties
```

Note, it will try to give suggestions about what the problem could be.

Grouper and UTF-8



utf8mb4 is not recommended because the database engines in MySQL/MariaDB that support transactions have limitations on the length of prefixes for indices. For InnoDB and XtraDB, the prefix is limited to 767 bytes. When using utf8mb4, where 4 bytes are used instead of 2, the prefixes used for some Grouper tables are too long.

For Grouper to handle UTF-8 characters:

- Make sure java starts with this setting: `-Dfile.encoding=UTF-8` for example, in tomcat, put something like this in your startup script:

```
export JAVA_OPTS="-server -Xms50M -Xmx200M -XX:MaxPermSize=95M -Dfile.encoding=UTF-8"
```

- Make sure URIEncoding is set to UTF-8 in the server.xml in tomcat in the Connector element, here is an example

```
<Connector port="8111" protocol="AJP/1.3" request.tomcatAuthentication="false"
    URIEncoding="UTF-8" />
```

- Make sure your database tables are using UTF-8. For Mysql you might want to use [utf8mb4](#). If not, e.g. for MySQL, you might need to SQL dump, delete the DB, create with default UTF-8 bin collation, and import the SQL.
- Make sure your connect string to the database is configured to use UTF-8 (if applicable). For MySQL, here is an example (see the stuff after the question mark). Note, in Windows MySQL this isn't necessary

```
hibernate.connection.url = jdbc:mysql://localhost:3306/grouper_v2_2?
Charset=utf8&useUnicode=true&characterEncoding=utf8
```

Grouper v2.2+ translations

If you want to add a localization to the Grouper UI text (example for French):

Add this to your grouper.properties (index must increment from the base file):

```
# language for this bundle
grouper.text.bundle.1.language = fr

# country for this bundle
grouper.text.bundle.1.country = fr

# filename in the package grouperText that is before the .base.properties, and .properties
grouper.text.bundle.1.fileNamePrefix = grouperText/grouper.text.fr.fr
```

Add a grouperText/grouper.text.fr.fr.base.properties and grouper.text.fr.fr.properties

Have this at the top of grouper.text.fr.fr.base.properties

```
#####
## Config chaining hierarchy
#####

# comma separated config files that override each other (files on the right override the left)
# each should start with file: or classpath:
# e.g. classpath:grouperText/grouper.text.en.us.base.properties, file:c:/temp/grouperText/grouper.text.en.us.
properties
text.config.hierarchy = classpath:grouperText/grouper.text.fr.fr.base.properties, classpath:grouperText/grouper.
text.fr.fr.properties
```

Change your browser to prefer French from France, (either wait a bit or bounce app server), and hit the UI and you should see your French text

Code changes

In the tomcat server.xml Edit the <Connectors to have uri encoding of utf8, e.g. <Connector URIEncoding="UTF-8"

edu.internet2.middleware.grouper.ui.util.HttpContentType.java

FROM:

```
TEXT_XML("text/xml"),

/** text html content type */
TEXT_HTML("text/html"),

/** application json content type */
APPLICATION_JSON("application/json"),
```

TO:

```
TEXT_XML("text/xml;charset=utf-8"),

/** text html content type */
TEXT_HTML("text/html"),

/** application json content type */
APPLICATION_JSON("application/json;charset=utf-8"),
```

Insert some subjects in the DB/LDAP (if you don't already have some in an existing source, note, you need the default source enabled if you use this sql):

```
insert into subject (subjectId, subjectTypeId, name) values ("joñ", "person", "joñ");

insert into subjectattribute (subjectId, name, value, searchValue) values ("joñ", "description", "Joñ", "joñ");
insert into subjectattribute (subjectId, name, value, searchValue) values ("joñ", "loginid", "joñ", "joñ");
insert into subjectattribute (subjectId, name, value, searchValue) values ("joñ", "name", "Joñ", "joñ");
commit;
```

grouper/grouper-ui.base.properties:

```
#####
## Internationalization
#####

# as of 2014/09/01 and Grouper 2.2.1+, this should be false
convertInputToUtf8 = false
```

edu.internet2.middleware.grouper.j2ee.GrouperRequestWrapper:

FROM:

```
public String getParameter(String name) {  
    if (!this.multipart) {  
        return this.wrapped.getParameter(name);  
    }  
  
    Object objectSubmitted = this.parameterMap.get(name);
```

TO:

```
public String getParameter(String name) {  
    if (!this.multipart) {  
        String param = this.wrapped.getParameter(name);  
        if (param != null && StringUtils.equals("GET", this.getMethod()) && TagUtils.mediaResourceBoolean  
("convertInputToUtf8", true)) {  
            try {  
                byte[] bytes = param.getBytes("ISO-8859-1");  
                param = new String(bytes, "UTF-8");  
            } catch (Exception e) {  
                throw new RuntimeException(e);  
            }  
        }  
        return param;  
    }  
  
    Object objectSubmitted = this.parameterMap.get(name);
```

edu.internet2.middleware.grouper.ui.GrouperUiFilter:

FROM:

```
public void doFilter(ServletRequest servletRequest, ServletResponse response,  
    FilterChain filterChain) throws IOException, ServletException {  
  
    GrouperRequestWrapper httpServletRequest = null;  
  
    try {  
        httpServletRequest = new GrouperRequestWrapper((HttpServletRequest) servletRequest);
```

TO:

```
public void doFilter(ServletRequest servletRequest, ServletResponse response,  
    FilterChain filterChain) throws IOException, ServletException {  
  
    GrouperRequestWrapper httpServletRequest = null;  
  
    try {  
        servletRequest.setCharacterEncoding("UTF-8");  
  
        httpServletRequest = new GrouperRequestWrapper((HttpServletRequest) servletRequest);
```

sdf

Testing

- Create folder on new UI with special char
- Create group in that folder with special char

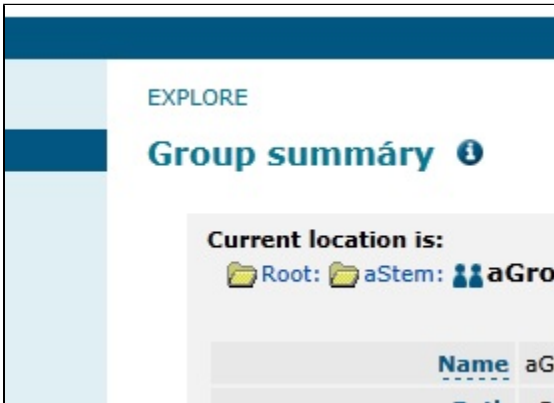
- Click on the folder breadcrumb to test is
- Search for the folder in a folder combobox, select it
- Put accented chars in the grouper text file and see those on the UI
- Do the tests below too

Text on admin UI

custom/nav.properties

```
groups.action.show-summary=Group summáry
```

Looks like this on Group screen:



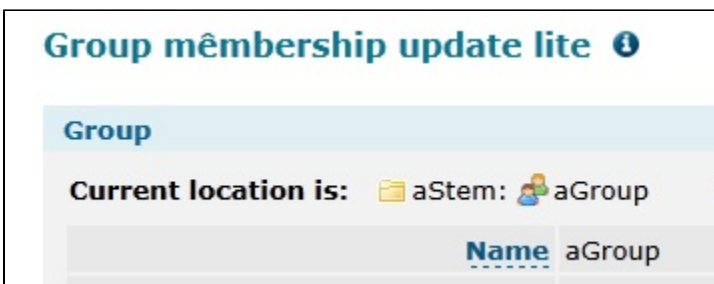
Text on Lite UI

Set this on custom/media.properties

```
ui-lite.link-from-admin-ui = true
login.ui-lite.show-link = true
```

custom/nav.properties

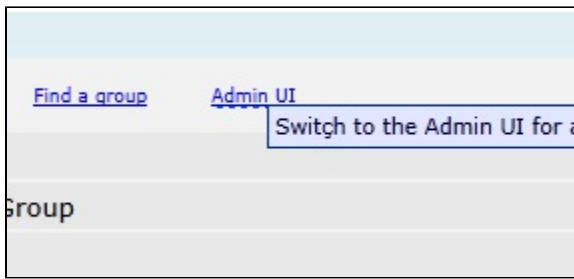
```
simpleMembershipUpdate.updateTitle=Group mêmbership update lite
```



Tooltip on the lite UI

custom/nav.properties

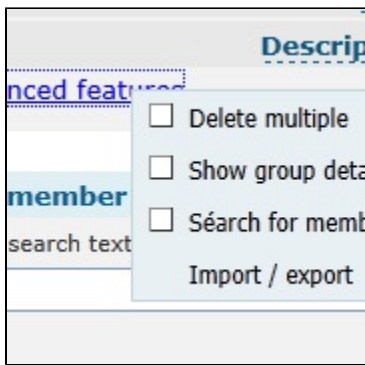
```
tooltipTargetted.simpleMembershipUpdate.viewInAdminUi=Switch to the Admin UI for a more complete set of features
```



Menu item lite

custom/nav.properties

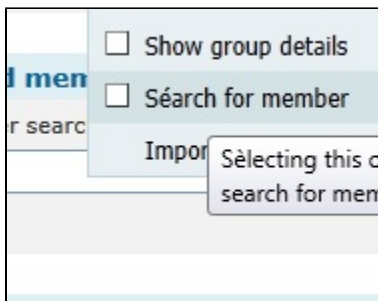
```
simpleMembershipUpdate.advancedMenuShowMemberFilter=Séarch for member
```



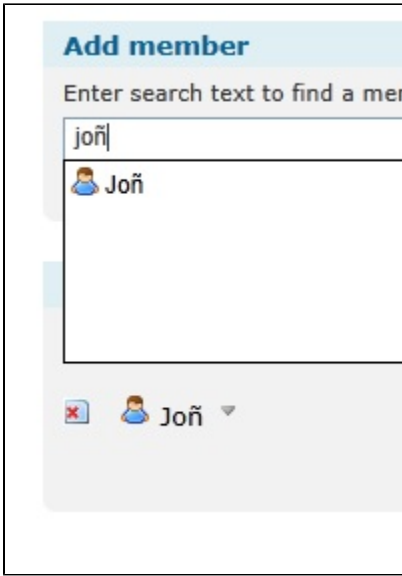
Menu tooltip lite

custom/nav.properties

simpleMembershipUpdate.advancedMenuShowMemberFilterTooltip=Sélectionner cette option affichera une zone de recherche au-dessus de la liste des membres où vous pouvez rechercher des membres dans ce groupe



Lite combo and submitting



See Also

[Grouper Configuration Overlays](#)

[API Building and Configuration](#)
