Customization

To customize clowder you can put all configuration changes in a folder called custom inside the clowder folder. If you are working on the source code this folder is excluded from git so you can use that also to customize your development environment, and not accidentally commit changes to either play.plugins or application.conf. If you make any changes to the files in the custom folder you will need to restart the application (both in production and development).

play.plugins

The play.plugins file is used to enable plugins. You can only enable plugins, you can not disable plugins. This is one of the reasons why we minimize the number plugins that are enabled by default. For example most instances at NCSA will have the following plugins enabled.

```
9992:services.RabbitmqPlugin
11002:services.ElasticsearchPlugin
```

custom.conf

The custom.conf file is used to override any of the changes in the application.conf or any included conf files (such as securesocial.conf). One change every instance of clowder should do is to modify the commKey and application.secret. Common changes we do is to modify clowder to use a directory on disk to store all blobs instead of storing them in mongo. Following is an example that we use for some of the instances we have at NCSA.

```
# security options
application.secret="12345678901234567890123456789012345678901234567890"
commKey=notreallyit

# email when new user tries to sign up
smtp.from="no-reply@example.com"
smtp.fromName="NO REPLY"

# URL to mongo
mongodbURI = "mongodb://mongo1:27017,mongo2:27017,mongo3:27017/server1?replicaSet=CLOWDER"

# where to store the blobs
service.byteStorage=services.filesystem.DiskByteStorageService
medici2.diskStorage.path="/home/clowder/data"

# rabbitmq
clowder.rabbitmq.uri="amqp://user:password@rabbitmq/clowder"
clowder.rabbitmq.exchange=server1

initialAdmins="joe@example.com"

# elasticsearch
elasticsearchSettings.clusterName="clowder"
elasticsearchSettings.serverAddress="localhost"
elasticsearchSettings.serverPort=9300

# securesocial customization
securesocial.ssl=true
securesocial.cookie.idleTimeoutInMinutes=480

# twitter setup
securesocial.twitter.requestTokenUrl="https://api.twitter.com/oauth/request_token"
securesocial.twitter.accessTokenUrl="https://api.twitter.com/oauth/access_token"
securesocial.twitter.authorizationUrl="https://api.twitter.com/oauth/authorize"
securesocial.twitter.consumerKey="key"
securesocial.twitter.consumerSecret="secret"

# enable cache
ehcacheplugin = enabled
```

messages.XY

This allows to translate or customize certain aspects of clowder. All messages in clowder are in english and are as messages.default. Unfortunatly it is not possible to use messages.default to use for translations since it falls back to those embedded in the clowder jar files. To update the messages in english, you can use messages.en. The default is for clowder to only know about english, this can be changed in your custom.conf with `application.langs="nl"`.

public folder

The public folder is place where you can place customizations for previews, as well as new stylesheets. To add a new stylesheet you should place it in the `public/stylesheets/themes/` folder. The name should be <something>.min.css or <something>.css. The user will at this point see in their customization settings the option to select <something> as their new theme to be used.

To add new previews you can put them in the `public/javascripts/previewers/`. To create a previewer you will create a folder in there and in there have the files needed for the previewer as well as a package.json file. This package.json file will describe the previewer, which as the name, the main file to load, and the content types (Preview files) that the previewer can handle.

```
{ "name" : "Video",
  "main" : "video.js",
  "contentType" : ["video/webm", "video/mp4", "video/videoalternativeslist"]
}
```