

## **LionShare Peer Customization Guide DRAFT**

Alex Valentine

[alexvalentine@psu.edu](mailto:alexvalentine@psu.edu)

### **Part 1: Summary and Requirements**

This document is a quick HOWTO for customizing the LionShare Peer for campus distribution. There are typically two options for campus installations used to test and evaluate the LionShare Peer.

#### **Option #1: Campus Service Integration**

The campus service integration option makes use of organizational authentication and directory services. The upside of a campus service integration approach is that LionShare uses official directory and identity information. The only drawback is that the group integrating LionShare on their campus cannot create accounts for users outside their official campus authentication realm.

#### **Option #2: Provide Your Own Infrastructure**

There are many situations which call for providing your own services for use with LionShare. A common request is that the group implementing LionShare wants the ability to create accounts for users outside of their institution. If this is the case, building authentication and directory infrastructure will permit the creation of outside user accounts. Another situation which calls for the build your own approach is when the campus does not provide adequate services for use with the LionShare peer.

There are many drawbacks with the build your own approach, the level of trust is clearly not as high as it would be with the use of campus infrastructure. The additional services will add additional tasks for the person in charge of administration. Finally, unless scripted, the management of user accounts requires an administrator to manually input account information.

#### **Infrastructure Requirements**

This guide assumes that the following infrastructure is already setup:

- Authentication Service
  - Either through campus services or set up particularly for LionShare.
  - Currently, the certificate authority component of LionShare (SASL-CA) only supports Kerberos authentication. Support is planned for additional authentication methods such as LDAP. Check the SASL-CA README file for latest authentication methods supported.
- Directory
  - LionShare can support a variety of directory services through use of the Shibboleth project's AttributeResolver code. The most common directory used with LionShare is LDAP. Typically, institutions doing a test integration of the LionShare Peer will use their campus's directory service with LionShare. However; another option is to setup a directory service. The most common directory server used is OpenLDAP, but others will work just fine too.
- Certificate Authority

- The core component for the LionShare security strategy is the SASL-CA. Certificates provided by the SASL-CA are the basis for LionShare transactions. This guide assumes that the reader has already setup and configured a SASL-CA. If a SASL-CA is not online, please view the README file included in the SASL-CA distribution.

SASL-CA and other LionShare downloads: <http://lionshare.its.psu.edu/downloads/>

## Build Requirements

Building the LionShare peer for your own distribution requires a proper system environment.

- Java SDK version 1.5.0 or greater
  - <http://java.sun.com>
- Apache ANT
  - <http://ant.apache.org/>

## Part 2: Security Configuration

### Step 1: Access and build source code

The easiest way to access the LionShare source code is to download the latest source tarball at: <http://lionshare.its.psu.edu/downloads/releases/lionshare-peer/source/>.

Extract the tarball `tar jxvf tarball-name.tar.bz2` in the command line or use a graphical utility.

Before attempting to customize the lionshare-peer, test the compilation environment

```
cd core
ant
cd ../gui
ant
```

If no errors appear, move on to step 2. If an environmental error is displayed, double check the Java and ANT configuration. For further assistance, e-mail [support@lionshare.its.psu.edu](mailto:support@lionshare.its.psu.edu) for help.

### Step 2: Security configuration

After the build environment is set, the main configuration change will be to point the LionShare Peer towards the SASL-CA and Kerberos realm. To do so, edit and recompile the following file:

```
core/edu/psu/its/lionshare/settings/LionShareApplicationSettings.java
```

Starting on line 41, you see the following settings:

```
public static final StringSetting JAVA_SECURITY_KRB5_REALM =
    FACTORY.createStringSetting("JAVA_SECURITY_KRB5_REALM", "dce.psu.edu");

public static final StringSetting JAVA_SECURITY_KRB5_KDC =
    FACTORY.createStringSetting("JAVA_SECURITY_KRB5_KDC", "sherlock.aset.psu.edu");

public static final StringSetting JAVA_SECURITY_SASL_CA_IP =
```

```
FACTORY.createStringSetting("JAVA_SECURITY_SASL_CA_IP",  
    "fadela.et-test.psu.edu");
```

```
public static final IntSetting JAVA_SECURITY_SASL_CA_PORT =  
    FACTORY.createIntSetting("JAVA_SECURITY_SASL_CA_PORT", 61455);
```

Change `JAVA_SECURITY_KRB5_REALM` to the appropriate realm in your Kerberos configuration. The values of `JAVA_SECURITY_KRB5_KDC`, `JAVA_SECURITY_SASL_CA_IP`, and `JAVA_SECURITY_SASL_CA_PORT` should also be set changed to point to your infrastructure. After saving, recompile for the changes to take effect.

```
cd core  
ant
```

### Step 3: Testing the Changes

After the successful compilation of the peer, the configuration changes should be tested by launching the application with the appropriate run script.

```
run.bat      Windows  
./runosx    OSX  
./run       Linux/Unix
```

Authenticate with a known account on the test realm. Authentication should be successful and the peer will automatically acquire certificates from the SASL-CA.

### Troubleshooting

*When I attempt to authenticate I get a username/password error?*

- Double check the settings for Kerberos in `LionShareApplicationSettings.java` (step1). Make sure the `JAVA_SECURITY_KRB5_REALM` and `JAVA_SECURITY_KRB5_KDC` settings are correct. If you need to make additional changes, be sure to recompile before testing again.
- Make sure that Kerberos is running on the server machine.
- Test your Kerberos realm by configuring an OS client and doing a manual kinit.
- Usernames and passwords are case sensitive, even when authenticating from Active Directory.

*Authentication is successful but the application times out on obtaining certificates from the SASL-CA?*

- Double check the settings for the SASL-CA in `LionShareApplicationSettings.java`, specifically `JAVA_SECURITY_SASL_CA_PORT` and `JAVA_SECURITY_SASL_CA_IP`.
- Confirm that the SASL-Ca is running on the server machine.
- If you are still experiencing problems, run the test client included with the SASL-CA.

### Part 3: Changing the Look and Feel

The look and feel of the LionShare Peer can be adjusted by individuals without any programming knowledge. There are a series of theme archives used by each platform LionShare supports. Each theme archive is located in `/lib/themes` directory. To change the look in feel for every supported platform, changes must be made to each specific theme archive.

<code>pinstripes_theme_osx.lwtp</code>	<i>OSX default theme file</i>
<code>gtk_theme_linux.lwtp</code>	<i>Linux default theme file</i>
<code>limewire_theme.lwtp</code>	<i>Windows default theme file</i>

Each theme archive file is a group of images and one configuration file enclosed in a zip file with the ending of `.lwtp`. To edit a theme, make a directory with any name and copy the theme file for editing. Extract the theme file using a zip utility or a command line tool.

Refreshing the directory list after extraction, a series of images and a `theme.txt` file should be displayed. The `theme.txt` file contains settings for colors used in the theme. After the desired changes to the `theme.txt` and images, rebuild the `lwtp` file for distribution. Using the command line this is accomplished by:

<code>rm limewire_theme.lwtp</code>	<i>Delete the old theme archive in the tmp directory</i>
<code>rm ../limewire_theme.lwtp</code>	<i>Delete the old theme archive in the themes directory</i>
<code>zip -r ../limewire_theme.lwtp *</code>	<i>Create a new theme zip file in the themes directory</i>

Once the new theme archive is created it will need to be included in a `theme.jar` file in two different locations. From the `lib/themes` directory:

```
jar -cvf ../jars/themes.jar *.lwtp
cp ../jars/themes.jar ../gui
h
```

To test the new theme, simply delete the `.lionshare` directory used in that particular platform, and execute the LionShare Peer. The new theme should be loaded by default, depending on the particular platform.

### Internationalization and Message Adjustments

The text components of the LionShare Peer are located in the `lib/messagebundles` directory. The default messagesbundle is `MessageBundle.properties`. To change any of the GUI text messages such as the application title or status messages, simple edit the properties file with any text editor.

For new language support, simple copy the default `MessageBundle.properties` and edit for the new language. The Limewire translation guide is a good starting point for learning more about internationalization.

Limewire Translation Guide  
<http://limewire.org/translate.shtml>

## Part 4: Distribution

LionShare is a cross platform application with native installers for Windows, OSX, and Linux/UNIX. To package a customized version for distribution, installers for each supported platform are a necessity. This guide covers how to make the most common installer for Windows 2000/XP. To create an installer for Windows, the maintainer will need to be working from the Windows 2000/XP platform.

The first step in the process is to install the Nullsoft Scriptable Install System or NSIS.  
NSIS [http://nsis.sourceforge.net/Main\\_Page](http://nsis.sourceforge.net/Main_Page)

After installing NSIS, compile the LionShare source code. From the root LionShare directory:

```
cd core
ant
cd ../gui
ant
```

At this point, deleting the .java source files is a good way to shave a few megabytes off the installer size. Use the windows search utility to find \*.java in the LionShare directory and delete any results found.

At this point, grab the NSIS configuration file (*lionshare.nsi*) from the util directory and place it in the root LionShare directory. Edit the *lionshare.nsi* file with a text editor. Any installer customizations should be made by editing the *lionshare.nsi* file. Once the customizations are finished, building the installer package is a simple process.

Launch the *makensisw* program from the start menu. Simply drag the *lionshare.nsi* file from the root directory of the LionShare Peer to the *makensisw* gui. The end result should be a windows EXE installer package in the root LionShare directory.

For information about packaging on other platforms, please e-mail [support@lionshare.its.psu.edu](mailto:support@lionshare.its.psu.edu).