KonaKart Portlet Installation for Liferay

9th January 2017

DS Data Systems (UK) Ltd.,
9 Little Meadow
Loughton, Milton Keynes
Bucks
MK5 8EH
UK

Table of Contents

KonaKart Portlets

Portlets can be created for both the storefront application and the admin application. They consist of WAR files that can be imported into Liferay.

Note that the default single sign on (SSO) mechanisms for the storefront and admin applications are simple implementations that do not provide the required security for a production environment. They need to be enhanced depending on the authentication system you are using with Liferay (CAS, OpenSSO etc.). Details are provided further down in the Single Sign On section of this document.

Supported Versions of Liferay

Only Liferay versions 6 & 7 are currently supported.

Install KonaKart

In order to create the Liferay portlets you need to first install KonaKart in the normal fashion. Follow the instructions in the User Guide for this purpose. (Typically this will simply involve running the relevant KonaKart installer(s)).

Create the WAR file

Create a command line window and navigate to the custom directory under your KonaKart installation:

On Windows:

```
C:\Users\Fred> cd "c:\KonaKart\custom"
```

On Linux:

```
fred@luton:~$ cd /home/fred/konakart/custom/
```

Note that ANT is provided in the download package which is sufficient for building the KonaKart WAR files for Liferay.

You can check the commands available to you using “**kkant -p**”, for example:

```
C:\KonaKart\custom>.\bin\kkant -p | findstr liferay
 make_admin_liferay_portlet_war  Create the konakartadmin portlet war for Liferay
 make_liferay_portlet_war        Create the konakart portlet war for Liferay
```

Note above there are two main Liferay targets. One is for the application (**make_liferay_portlet_war**) and one for the Admin Application (**make_admin_liferay_portlet_war**).

For example, to create the konakart.war for Liferay choose the **make_liferay_portlet_war** ANT target as follows:

```
C:\KonaKart\custom>.\bin\kkant make_liferay_portlet_war -DLR606=true
                                                                     -DnoAXIS=true
Buildfile: build.xml

clean_portlet_war:
     [echo] Cleanup portlet WARs...
     [echo] Cleanup portlet classes...
     [echo] Cleanup portlet WAR staging area...

make_liferay_portlet_war:
    [mkdir] Created dir:
        C:\KonaKart\custom\portlet_war
    [mkdir] Created dir:
        C:\KonaKart\custom\konakart_portlet\stage\konakart
    [mkdir] Created dir:
        C:\KonaKart\custom\konakart_portlet\stage\konakart\WEB-INF\jsp
     [echo] Lay out the WAR in the staging area
     [echo] Copy (almost) the whole konakart webapp to staging area
     [copy] Copying 834 files to
        C:\KonaKart\custom\konakart_portlet\stage\konakart
     [copy] Copied 35 empty directories to 1 empty directory under
        C:\KonaKart\custom\konakart_portlet\stage\konakart
     [echo] Copy the jars reqd for the portlet to staging area
     [copy] Copying 2 files to
        C:\KonaKart\custom\konakart_portlet\stage\konakart\WEB-INF\lib
     [echo] Copy the config files reqd for the portlet to staging area
     [copy] Copying 6 files to
        C:\KonaKart\custom\konakart_portlet\stage\konakart\WEB-INF
     [echo] Filter the JSPs to staging area for the portlet WAR
     [echo] Filter the web.xml to staging area for the portlet WAR
     [echo] Filter the struts-config.xml to staging area for the portlet WAR

adjustAppPortletForLR605:

adjustAppPortletForLR606:
     [echo] Make adjustments to Application Portlet WAR for Liferay 6.0.6
     [echo] Copy the IterateTag patch to staging area
     [copy] Copying 1 file to
        C:\KonaKart\custom\konakart_portlet\stage\konakart\WEB-INF\classes

adjustAppPortletForLR611:

adjustAppPortletForAXIS:
     [echo] Make adjustments to Application Portlet WAR to exclude AXIS

adjustAppPortletForJBoss:

adjustAppPortletForJBossLiferay:

adjustAppPortletForJBossLiferay606:

adjustAppPortletForJBossLiferay611:

adjustAppPortletForKKDemoSite:
     [echo] Create portlet konakart.war
      [war] Building war: C:\KonaKart\custom\portlet_war\konakart.war

BUILD SUCCESSFUL
```

Do not be concerned that your build matches the number of files precisely because this can change depending on your version and configuration.

For convenience some common adjustments to the jars that are included can be made by setting the **-DLRnnn** parameter for the applicable Liferay version.   Examples are -**DLR605**, -**DLR606**, -**DLR611, -DLR6120** (for Liferay 6.1.20), -**DLR62, -DLR7**.

For other versions of Liferay it is possible that other adjustments will be required.

Note for **Jboss/Liferay** bundle users:

Add the "**-Djbossliferay620=true**" argument as follows.   (note that the "**jbossliferay620**" parameter must be lowercase).   (Check the custom/build.xml file for other supported version options such as

-Djbossliferay6120).

```
/home/brian/konakart/custom>./bin/kkant make_admin_liferay_portlet_war
                            -Djbossliferay620=true
                            -DLR620=true
                            -DnoAXIS=true
```

When creating a portlet WAR for the **Jboss/Liferay 6.0.6 bundle** add the "**-Djbossliferay606=true**" argument as follows:

```
/home/brian/konakart/custom>./bin/kkant make_admin_liferay_portlet_war
                            -Djbossliferay606=true
                            -DLR606=true
                            -DnoAXIS=true
```

Enter all the commands above on one line only.

Note the various "**adjustForJBossLiferay**" targets in the build.xml file, for example:

```
<target name="adjustAdminPortletForJBossLiferay" if="forJBbossLiferay">
    <echo message="Make adjustments for JBoss-Liferay" />
    <delete failonerror="true">
        <fileset
            dir="${custom.home}/konakartadmin_portlet/stage/konakartadmin/WEB-INF/lib">
      <include name="xercesImpl-*.jar" />
      <include name="xml-apis-*.jar" />
      <include name="portal-kernel.jar" />
      <include name="portal-service.jar" />
      <include name="portlet.jar" />
      <include name="activation.jar" />
      <include name="mail.jar" />
      <include name="quartz-*.jar" />
      <include name="commons-logging-*.jar" />

      <!-- Adjust these to suit your own environment -->
      <include name="db2*.jar" />
      <include name="jtds*.jar" />
      <include name="postgre*.jar" />
      <include name="ojdbc14.jar" />
        </fileset>
    </delete>
</target>
```

You can see that the default case is to exclude the jars for DB2, PostgreSQL, Oracle, MS-SQL Server... You may need to adjust this to suit your own environment.   Normally all of the jars are included but sometimes deploying very large WARs with many JARs into Jboss gives intermittent ZipExceptions hence it is advisable to cut down the number of JARs in the WAR.

Import the WAR into Liferay

The konakart / konakartadmin war files can be imported into Liferay by copying them to the Liferay deploy directory.

Once they have been imported you need to add them to a Liferay page. We've found that the Liferay 1 column template for the new page works well. Depending on your Liferay theme you may have to make some KonaKart style changes in order to display the portlets correctly.

When integrating with Liferay 7, the time taken to deploy the war files can be many minutes, so please be patient!

General Notes

Problems with AXIS

Some portal systems use a version of AXIS that conflicts with the one in KonaKart. A workaround for this problem is to disable the AXIS web services in your portlet (if you need the web services you can always run them in a servlet container on another machine where the instance doesn't run as a portlet).

A flag can be added to the ANT command to eliminate the WSDD files which in turn will disable the AXIS web services and stop the start-up exception that you get in systems where there is an incompatible version of AXIS.
The flag to use to disable AXIS web services is "-**DnoAXIS**=true" hence your ANT command line would be:

```
C:\KonaKart\custom>.\bin\ant make_admin_liferay_portlet_war -DnoAXIS=true -DLR620
```

A similar conflict exists for the JAXWS interfaces. By default these are disabled in portlets created for Liferay.

IE 8 users running inside Liferay:

Note that this should no longer be a problem from Liferay 6.0.5 and above.

The Admin App does not currently support IE 8 so you either have to run the browser in compatibility mode or enter the following meta information in the head tag of your theme:

```
 <meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7" />
```

Theme files are typically located under deploy/ROOT.war/html/themes/ An example of a theme file under this directory is classic/templates/portal_normal.vm

Problems Installing the portlets in Liferay

Because of the various combinations and permutations of jars used in the various Liferay installation kits (tomcat/JBoss/etc of various versions plus various versions of Liferay itself) and also because we don't know which jars have been placed in common lib directories in these installations we cannot create a perfect WAR for every case. The WARs that are created are a good start – and work fine in most cases – but you may find the installation does not complete successfully.

If your installation fails a common problem is that the jars that Liferay has in its common libs clash with those in the konakart or konakartadmin webapps. In most cases the solution is simply to remove the clashing jars from the konakart or konakartadmin webapp lib directory, recreate the WAR and re-deploy.

Different behavior is exhibited with different Liferay bundles. It was found that with the standard konakart / konakartadmin portlet war produced in the KonaKart v5.2.0.0 system, it installed successfully in Liferay 6.0.5 when bundled with either Glassfish, JBoss or Resin but not with Tomcat, Jetty or Geronimo.

For Liferay 6.0.5 + Tomcat 6.0.26 and Liferay 6.0.6 + Tomcat 6.0.29 you must create your konakartadmin portlet war after removing the following jars:

```
                         WEB-INF/lib/portal-kernel.jar
WEB-INF/lib/portal-service.jar
WEB-INF/lib/portlet.jar
```

All known problems with using KonaKart portlets on Liferay 6.0.5/Tomcat 6.0.26 and Liferay 6.0.6/Tomcat 6.0.29 such as these can be solved by building the portlet WARs using the "**-DLR605=true**" option (see example below) or the "**-DLR606=true**" option appropriate for your version of Liferay:

```
C:\KonaKart\custom>.\bin\ant make_liferay_portlet_war
                             make_admin_liferay_portlet_war
                             -DnoAXIS=true
                             -DLR605=true
```

Enter all the commands above on one line only.

Don't forget your Memory.

It is common to need to increase the default memory settings for Liferay (eg. -Xmx1200m -XX:MaxPermSize=512m). What you need will depend on your application server choice and what other applications/portlets are being run.

If in doubt, restart Liferay.

In practice it has been observed that on certain occasions it is necessary to restart Liferay after a deployment of the KonaKart portlets for these to become available for use.Single Sign On.

Resource bundle not found ?   MySQL on Liferay DXP ?

In the default Liferay DXP (v7) bundle we experienced problems when using KonaKart with MySQL probably caused by classloading problems and an inability to locate MySQL Localised Error Messages for the current locale.   To overcome this problem we moved our mysql JDBC jar (the one in the webapps/konakart/WEB-INF/lib directory in KonaKart) into the Liferay tomcat lib/ext directory.

Summary of steps to take for Liferay DXP

As an example, these are the steps to take when using Liferay DXP.

1. When building the WARs use the flags -DLR7=true -DnoAXIS=true
2. If using the MySQL database, copy the KonaKart MySQL jar from the webapps/konakart/WEB-INF/lib directory in KonaKart to the Liferay tomcat lib/ext directory if it doesn't already exist in this directory.

KonaKart Admin Portlet

Setting Roles

Liferay users are assigned KonaKart Admin App roles based on the names of the Liferay roles. The Liferay role also contains the store id encoded within the role.

Example Liferay roles:

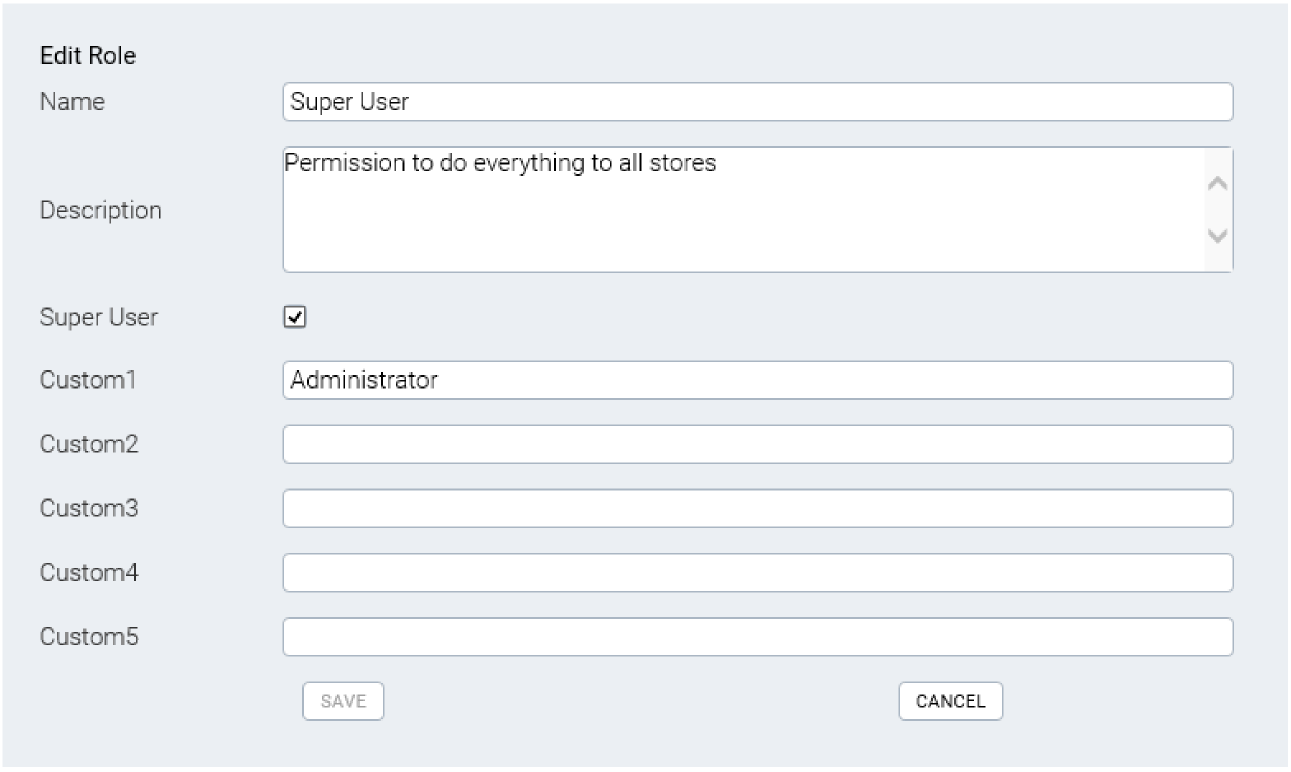
- administrator
- catalog_store2
- order_store1

catalog_store2	Regular	Maps to KonaKart catalog for store2	Actions

In order for KonaKart to recognize the above roles, the custom1 attribute of the matching KonaKart role must contain the name of the Liferay roles (without the store id), which in this case would be:

- administrator
- catalog
- order

Note that in this case the administrator Liferay role does not have a store id because it maps to the administrator KonaKart role which gives Super-User privileges so that the user can access all stores.



Edit Role

Name	Super User
Description	Permission to do everything to all stores
Super User	☑
Custom1	Administrator
Custom2	
Custom3	
Custom4	
Custom5	

Note that you won't be able to use the Liferay-based Admin App portlet until you set the role names in the Roles... So there is chicken-and-egg problem here!

There are two alternative solutions....   Either:

1. Run the tomcat-based KonaKart Admin App that you have installed that points to the same database and edit the roles there.
2. Update the custom1 column in the kk_role table manually using your favorite database query tool.

Note that from KonaKart v5.2.0.0 the Super User role has the Custom1 column set to “Administrator” for convenience.

SSO Code

The source of the code that performs the SSO is only available in the Enterprise version of KonaKart.

The name of the java class that performs the role matching is KKAdminPortlet.java and can be found after a default installation in the location:

custom\konakartadmin_portlet\liferay\src\com\konakartadmin\portlet\

If a KonaKart user matching the email address of the Liferay user, doesn't yet exist, then the user is registered. Otherwise the roles of the KonaKart user are deleted and then added again just in case they changed since the last login.

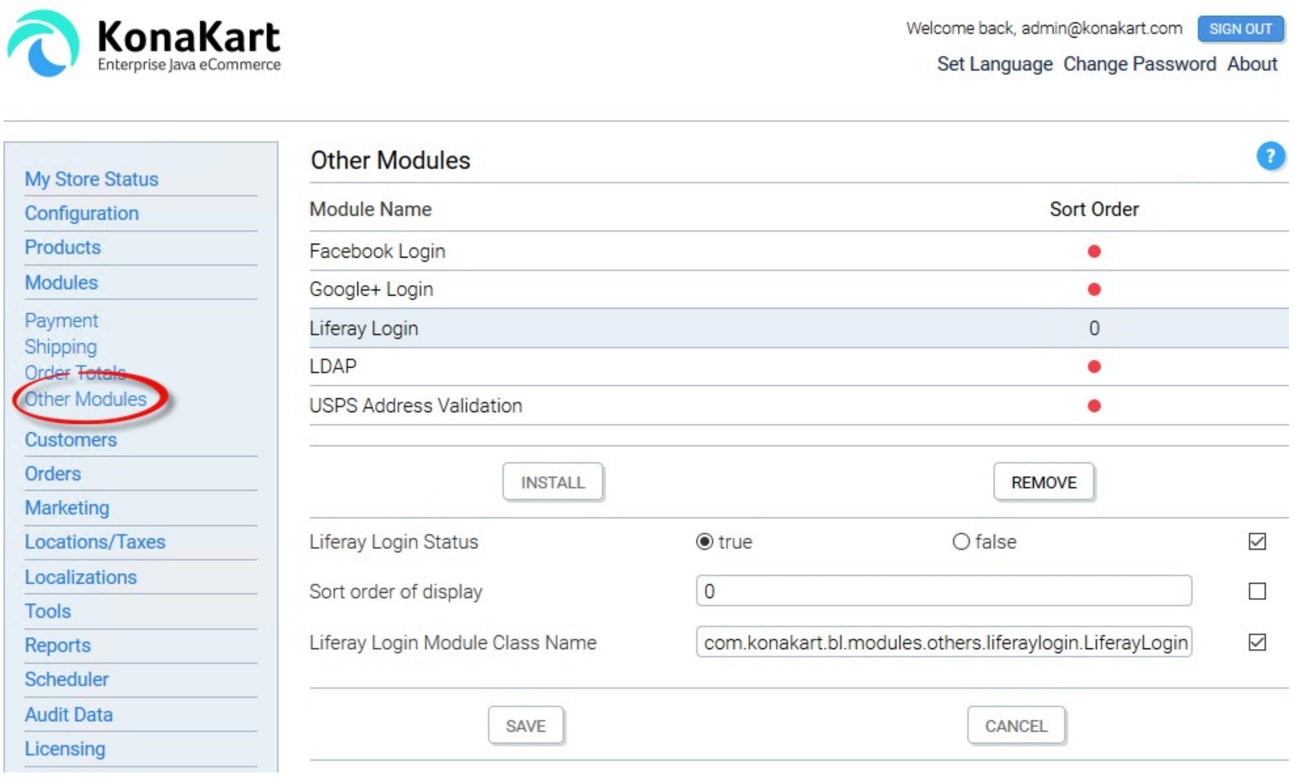
The AdminLoginIntegrationMgr can be found under:
custom\konakartadmin_portlet\liferay\src\com\konakartadmin\bl

It is a modified version that doesn't perform credential checking.

Note that the customization made in the AdminLoginIntegrationMgr (so that the admin user can be logged in automatically) overrides credential checking and so if the Admin App is available directly from a URL rather than through Liferay (or from APIs) then all security will be disabled. In order to implement tighter security, you should use an SSO system and pass a token through to this method so that it can check with the SSO service whether the token is valid.

KonaKart Storefront Application Portlet

To achieve SSO with the storefront, the first step is to install the Liferay Login module from the admin app as shown below. Note that this is only available in the Enterprise version of KonaKart.



Other Modules

Module Name	Sort Order
Facebook Login	●
Google+ Login	●
Liferay Login	0
LDAP	●
USPS Address Validation	●

Liferay Login Status	[✓ true]	false	☑
Sort order of display	0		☐
Liferay Login Module Class Name	com.konakart.bl.modules.others.liferaylogin.LiferayLogin		☑

The default implementation of the module logs in the Liferay customer when called through the API call:

public ExternalLoginResultIf externalLogin(ExternalLoginInputIf loginInfo) throws KKException

The client side part of the SSO code resides in SingleSignOn.java which can be found in
custom\konakart_portlet\liferay_sso\src\com\konakart\liferay

after a default installation. Here, customer information is retrieved from Liferay and the customer is registered with KonaKart (if the customer doesn't already exist) and then logged in by calling the Liferay Login Module.

When using this default implementation, you must ensure that the externalLogin() API call isn't available through SOAP, JSON or RMI accessible from outside of the firewall. The reason is that the default implementation of the module doesn't make any server side checks to ensure that the customer is logged in. The module would need to be enhanced to make such a check if you are using authentication systems such as CAS or OpenSSO.