

Liferay Portal 4 - Customization Guide

Joseph Shum

Alexander Chow

Jorge Ferrer

Liferay Portal 4 - Customization Guide

Joseph Shum
Alexander Chow
Jorge Ferrer

1.0

Copyright © 2000, 2007 Liferay Inc.

Table of Contents

Preface	v
1. Configuration Options	1
1. Customizing the configuration	1
2. portal.properties	1
3. system.properties	42
2. Customizing Liferay with plugins	47
1. Introduction to Portlets	47
2. Introduction to Themes	47
3. Introduction to Layout Templates	49
3. Deployment of Plugins	50
1. Deployment Options	50
1.1. Standard Deployment	50
1.2. Hot Deployment	50
2. Deploying Third Party plugins	50
2.1. Deploying Community Themes	51
2.2. Adding Layout Template	54

Preface

Intended audience. This document is intended as a guide for those who have already installed Liferay Portal and want to customize it for specific needs. It covers configuration options and deployment of existing themes, portlets, and layouts. It does not cover development of new themes, portlets, or any other plugins.

Liferay version. This guide has been written for Liferay 4. Some details might be different for previous versions. Do not expect it to be accurate for older versions.

Related documents. If this is not what you are looking for consider the following related documents:

- Liferay Portal 4 - Installation Guide
- Liferay Portal 4 - Portal Administration Guide
- Liferay Portal 4 - Development in the Extension Environment

More information and support. If you have are looking for help for an specific issue we invite you to use our community forums: <http://www.liferay.com/web/guest/devzone/forums> [<http://forums.liferay.com>] to ask your questions. We also offer professional support services (support@liferay.com [<mailto:support@liferay.com>]) where your company will be assigned a Liferay developer ensuring your questions are answered promptly so that your project is never compromised. Purchased support always gets first priority. This business model allows us to build a company that can contribute a great portal to the open source community. If your company uses Liferay, please consider purchasing support. Liferay has an extremely liberal license model (MIT, very similar to Apache and BSD), which means you can rebundle Liferay, rename it, and sell it under your name. We believe free means you can do whatever you want with it. Our only source of revenue is from professional support and consulting.

Chapter 1. Configuration Options

Liferay has been built to be highly customizable through the web and through configuration files. This chapter reviews the two files where Liferay Portal stores its configuration, `portal.properties` and `system.properties`, and how to override their values.

1. Customizing the configuration

Liferay provides an easy way to override the properties of its configuration files. This can be done through the files `portal-ext.properties` and `system-ext.properties` that can be created and stored in any place in the classpath. When the Liferay extension development environment is used these files are already present in the directory `ext-ejb/WEB-INF/classes`. Otherwise, it is recommended to store them in the global classpath of the application server. For example:

- Tomcat: place them in `TOMCAT_DIR/shared/classes`
- JBoss: place them in `JBOSS_DIR/server/default/conf`
- Other application server: read the documentation provided with them.

Let's see an example of how to customize Liferay Portal's configuration. In this example Liferay Portal has been installed using the Tomcat bundle and then a custom theme called *mytheme* has been deployed as a WAR (using the procedure explained later in this document). The administrator wants this theme to be used by default in any newly created desktop or community. To achieve that he/she looks in `portal.properties` and finds that the property `default.theme.id` can be used to set the default theme and `default.color.scheme.id` the default color scheme of that theme. In this case the theme only has one color scheme called *mycolorscheme* so the administrator creates the file `portal-ext.properties` in `JBOSS_DIR/server/default/conf` with the following contents:

```
default.theme.id=mytheme
default.color.scheme.id=mycolorscheme
```

After a server reboot the new properties are applied to the portal. The following sections include an annotated copy of both `portal-ext.properties` and `system-ext.properties` that can be used as a reference of the available properties.

Note

Liferay uses EasyConf to read `portal.properties`, so all functionalities provided by this library are also available. You can read more about it in EasyConf's website [<http://easyconf.sourceforge.net>]

2. portal.properties

The main configuration file for Liferay Portal is `portal.properties`, which contains detailed explanation about the properties that it defines. To change the value of any of its properties do it through a file called `portal-ext.properties`.

```
##
## Properties Override
##

#
# Specify where to get the overridden properties. Updates should
```

```
not be made
  # on this file but on the overridden version of this file.
Furthermore, each
  # portal instance can have its own overridden property file
following the
  # convention portal-companyid.properties.
  #
  # For example, one read order may be: portal.properties, then
  # portal-ext.properties, then portal-liferay.com.properties.
  #
include-and-override=portal-ext.properties
include-and-override=portal-${easyconf:companyId}.properties
include-and-override=portal-test.properties

##
## Portal Release
##

  #
  # The enterprise release of the portal must run inside a J2EE 1.3
compliant
  # application server. The professional release of the portal can
run inside
  # a standalone servlet container.
  #
portal.release=enterprise
#portal.release=professional

##
## Portal Context
##

  #
  # Specify the location of the shared portal web context. All other
portals
  # utilize this shared instance of the portal web context.
  #
  # For example, the web application by the name of
"liferay.com-web" is
  # deployed with the context root at "/" while the web application
by the
  # name of "portal-web" is deployed with the context root at
"/portal". The
  # web application at "/" is lean and doesn't contain any actual
JSPs, but
  # uses the JSPs and resources found at "/portal".
  #
  # Another web application deployed with the context root at
"/another" could
  # also be its own portal instance by reusing the resources found
at
  # "/portal". This allows one server to host multiple instances of
the portal
  # while minimizing deployed resources.
  #
portal.ctx=/
portal.instances=1

##
## Error
##

  #
  # Set the following to true to log the error message.
```

```
#
error.message.log=true

#
# Set the following to true to print the error message to the
console.
#
error.message.print=false

#
# Set the following to true to show the error message to the user.
#
error.message.show=true

#
# Set the following to true to log the stack trace.
#
error.stack.trace.log=false

#
# Set the following to true to print the stack trace to the
console.
#
error.stack.trace.print=true

#
# Set the following to true to show the stack trace to the user.
#
error.stack.trace.show=false

##
## TCK
##

#
# Set the following to true to enable programmatic configuration
to let the
# Portlet TCK obtain a URL for each test. This should never be set
to true
# unless you are running the TCK tests.
#
tck.url=false

##
## Upgrade
##

#
# Input a list of comma delimited class names that implement
# com.liferay.portal.upgrade.UpgradeProcess. These classes will
run on
# startup to upgrade older data to match with the latest version.
#
upgrade.processes=com.liferay.portal.upgrade.UpgradeProcess_Legacy,com.liferay.por
com.liferay.portal.upgrade.UpgradeProcess_4_0_0,com.liferay.portal.upgrade.Upgrade
com.liferay.portal.upgrade.UpgradeProcess_4_2_0

##
## Auto Deploy
##

#
# Set the following to true to enable auto deploy of layout
templates,
```



```
# portlets, and themes.
#
auto.deploy.enabled=true

#
# Set the directory to scan for layout templates, portlets, and
themes to
# auto deploy.
#
auto.deploy.deploy.dir=/home/liferay/deploy

#
# Set the directory where auto deployed WARs are copied to. The
application
# server or servlet container must know to listen on that
directory.
# Different containers have different hot deploy paths. For
example, Tomcat
# listens on "../webapps" whereas JBoss listens on
# "../server/default/deploy".
#
auto.deploy.dest.dir=../webapps
#auto.deploy.dest.dir=../server/default/deploy
#auto.deploy.dest.dir=../../webapps/autoload

#
# Set the interval in milliseconds on how often to scan the
directory for
# changes.
#
auto.deploy.interval=10000

#
# Set the following to true if deployed WARs are unpacked.
#
auto.deploy.unpack.war=true

#
# Set the path to Tomcat's global class loader. This property is
only used
# by Tomcat in a standalone environment.
#
auto.deploy.tomcat.lib.dir=../common/lib/ext

##
## Resource Actions
##

#
# Input a list of comma delimited resource action configurations
that will
# be read from the class path.
#
resource.actions.configs=resource-actions/default.xml

##
## Model Hints
##

#
# Input a list of comma delimited model hints configurations.
#
model.hints.configs=META-INF/portal-model-hints.xml,META-INF/workflow-model-hints.
```

```
##
## Spring
##

#
# Input a list of comma delimited Spring configurations. The file
name will
# be namespaced depending on the release of the portal if and only
if the
# specified file name ends with spring.xml.
#
# For example, if you are starting Liferay Portal Professional,
then
# META-INF/counter-spring-professional.xml will be loaded. If you
are
# starting Liferay Portal Enterprise, then
# META-INF/counter-spring-enterprise.xml will be loaded.
#
# Certain configurations, like mail-spring-jms.xml, are shared by
both
# professional and enterprise editions and will not be
automatically
# namespaced because it does not end with spring.xml.
#
spring.configs=META-INF/counter-spring.xml,META-INF/documentlibrary-spring.xml,MET
META-INF/lock-spring.xml,META-INF/mail-spring.xml,META-INF/mail-spring-jms.xml,MET
META-INF/portal-spring-jcr.xml,META-INF/portal-spring-jms.xml,META-INF/ext-spring.

#
# Set the bean name for the Liferay data source.
#
spring.hibernate.data.source=liferayDataSource

#
# Set the bean name for the Liferay session factory.
#
spring.hibernate.session.factory=&liferaySessionFactory

##
## Hibernate
##

#
# Input a list of comma delimited Hibernate configurations.
#
hibernate.configs=META-INF/counter-hbm.xml,META-INF/mail-hbm.xml,META-INF/portal-h

#
# Use the Liferay SQL dialect because it will automatically detect
the proper
# SQL dialect based on your connection URL.
#
hibernate.dialect=com.liferay.portal.spring.hibernate.DynamicDialect

#
# Set the Hibernate cache provider. If you choose to use the
OSCache in a
# clustered environment, be sure to check with the OSCache
settings below.
#
#hibernate.cache.provider_class=org.hibernate.cache.EhCacheProvider
#hibernate.cache.provider_class=net.sf.hibernate.cache.HashtableCacheProvider
hibernate.cache.provider_class=com.liferay.portal.spring.hibernate.OSCacheProvider
```

```
#
# Set other Hibernate cache settings.
#
hibernate.cache.use_query_cache=true
hibernate.cache.use_second_level_cache=true
hibernate.cache.use_minimal_puts=true
hibernate.cache.use_structured_entries=false

#
# Uncomment these properties to disable Hibernate caching.
#
#hibernate.cache.provider_class=org.hibernate.cache.NoCacheProvider
#hibernate.cache.use_query_cache=false
#hibernate.cache.use_second_level_cache=false

#
# Set the JDBC batch size to improve performance. However, if
you're using
# Oracle 9i, you must set the batch size to 0 as a workaround for
a hanging
# bug in the Oracle driver. See
http://support.liferay.com/browse/LEP-1234
# for more information.
#
hibernate.jdbc.batch_size=20
#hibernate.jdbc.batch_size=0

#
# Set other miscellaneous Hibernate properties.
#
hibernate.jdbc.use_scrollable_resultset=true
hibernate.bytecode.use_reflection_optimizer=true
hibernate.show_sql=false

#
# Use the classic query factory until WebLogic and Hibernate 3 can
get
# along. See http://www.hibernate.org/250.html#A23 for more
information.
#
hibernate.query.factory_class=org.hibernate.hql.classic.ClassicQueryTranslatorFact

##
## Custom SQL
##

#
# Input a list of comma delimited custom SQL configurations.
#
custom.sql.configs=custom-sql/default.xml

#
# Some databases do not recognize a NULL IS NULL check. Set the
# "custom.sql.function.isnull" and "custom.sql.function.isnotnull"
# properties for your specific database.
#

#
# DB2
#
#custom.sql.function.isnull=CAST(? AS VARCHAR(32672)) IS NULL
#custom.sql.function.isnotnull=CAST(? AS VARCHAR(32672)) IS NOT
NULL
```

```
#
# MySQL (for testing only)
#
#custom.sql.function.isnull=ISNULL(?, '1') = '1'
#custom.sql.function.isnotnull=ISNULL(?, '1') = '0'

#
# Sybase
#
#custom.sql.function.isnull=IFNULL(?, '1') = '1'
#custom.sql.function.isnotnull=IFNULL(?, '1') = '0'

#
# Oracle has a nasty bug where it treats '' as a NULL value. See
# http://thedailywtf.com/forums/thread/26879.aspx for more
information on
# this nasty bug. Uncomment this if you're connecting to Oracle.
#
#custom.sql.vendor.oracle=true

##
## OSCache
##

#
# The following OSCache settings are used in conjunction with
Hibernate if
# you set Hibernate to use OSCache as its cache provider. Consult
the
# OSCache documentation for an exhaustive list of available
settings.
#
cache.memory=true
cache.capacity=10000
cache.algorithm=com.opensymphony.oscache.base.algorithm.LRUCache
cache.blocking=true

#
# Uncomment the "cache.event.listeners" property if you are
deployed in a
# clustered environment. Then uncomment either
"cache.cluster.multicast.ip"
# or "cache.cluster.properties", but do not uncomment both. You
must also
# configure the properties found in cache-multi-vm.properties to
safeguard
# your data in a cluster.
#
# The multicast ip is a unique namespace for a set of cached
objects.
# Set it to 231.12.21.100 to keep it unique from the multicast ip
set in
# cache-multi-vm.properties.
#
# Uncomment the "cache.cluster.properties" property if your cache
is binding
# to 127.0.0.1 and not to a network ip. Your cache must be
listening on a
# network ip to talk to other servers. You can test the clustered
cache by
# running two portals on two machines. Do not attempt to run both
portals on
# one machine because the multicast will not know to loopback.
#
```

```
#cache.event.listeners=com.opensymphony.oscache.plugins.clustersupport.JavaGroupsB
#cache.cluster.multicast.ip=231.12.21.100
#cache.cluster.properties=UDP(bind_addr=127.0.0.1;mcast_addr=231.12.21.100;mcast_p
ip_ttl=32;mcast_send_buf_size=150000;mcast_recv_buf_size=80000):PING(timeout=2000;
MERGE2(min_interval=5000;max_interval=10000):FD_SOCKET_VERIFY_SUSPECT(timeout=1500):
pbcast.NAKACK(gc_lag=50;retransmit_timeout=300,600,1200,2400,4800;max_xmit_size=81
UNICAST(timeout=300,600,1200,2400):pbcast.STABLE(desired_avg_gossip=20000):\
FRAG(frag_size=8096;down_thread=false;up_thread=false):\
pbcast.GMS(join_timeout=5000;join_retry_timeout=2000;shun=false;print_local_addr=t

##
## JavaScript
##

#
# Set the following to true to speed up the loading of JavaScript
files by
# loading /html/js/everything.js instead of loading multiple
files. See
# /html/common/themes/top_js.jsp and the Ant task build-javascript
found in
# portal-web for more information on how this works.
#
javascript.fast.load=false

#
# Set the following to true to enable the display of JavaScript
logging.
#
javascript.log.enabled=false

##
## Company
##

#
# The portal can authenticate users based on their email address
and
# password or based on their user id and password.
#
company.security.auth.type=emailAddress
#company.security.auth.type=userId

#
# Set the following to true to allow users to select the "remember
me"
# feature to automatically login to the portal.
#
company.security.auto.login=true

#
# Set the following to true to allow users to ask the portal to
send them
# their password.
#
company.security.send.password=true

#
# Set the following to true to allow strangers to create accounts
and
# register themselves on the portal.
#
company.security.strangers=false
```

```
##
## Users
##

#
# Set the following to false if users cannot be deleted.
#
users.delete=true

#
# Set the following to true to always autogenerate user ids even
if the user
# gives a specific user id.
#
users.id.always.autogenerate=false

#
# Input a class name that extends
# com.liferay.portal.security.auth.UserIdGenerator. This class
will be
# called to generate user ids.
#
users.id.generator=com.liferay.portal.security.auth.UserIdGenerator

#
# Input a class name that extends
# com.liferay.portal.security.auth.UserIdValidator. This class
will be
# called to validate user ids.
#
users.id.validator=com.liferay.portal.security.auth.UserIdValidator
#users.id.validator=com.liferay.portal.security.auth.LiberalUserIdValidator

#
# Set the maximum file size for user portraits. A value
# of 0 for the maximum file size can be used to indicate unlimited
file
# size.
#
users.image.max.size=307200

##
## Groups and Roles
##

#
# Input a list of comma delimited system group names that will
exist in
# addition to the standard system groups. When the server starts,
the portal
# checks to ensure all system groups exist. Any missing system
group will be
# created by the portal.
#
system.groups=

#
# Input a list of comma delimited system role names that will
exist in
# addition to the standard system roles. When the server starts,
the portal
# checks to ensure all system roles exist. Any missing system role
will be
# created by the portal.
```

```
#
system.roles=

#
# Omniadmin users can administer the portal's core functionality:
gc, # shutdown, etc. Multiple portal instances might be deployed on
one # application server, and not all of the administrators should
have access # to this core functionality. Input the ids of users who are
omniadmin # users. Leave this field blank if users with the Administrator
role are # allowed to administer the portal's core functionality.
#
omniadmin.users=liferay.com.1

#
# Set the following to true if all users can personalize pages. If
set to # false, only Administrators and Power Users can personalize.
#
universal.personalization=false

#
# Set the following to true if all users are required to agree to
the terms # of use.
#
terms.of.use.required=true

##
## Organizations and Locations
##

#
# Set the following to true if users must belong to a parent
organization. #
organizations.parent.organization.required=false

#
# Set the following to true if users must belong to a location. If
location # is required, then a parent organization is also required.
#
organizations.location.required=false

##
## Languages and Time Zones
##

#
# Specify the available locales. Messages corresponding to a
specific # language are specified in properties files with file names
matching that # of content/Language_*.properties. These values can also be
overridden in # properties files with file names matching that of
# content/Language-ext_*.properties. Use a comma to separate
# each entry.
#
```

```
# All locales must use UTF-8 encoding.
#
# See the following links specifiy language and country codes:
#   http://ftp.ics.uci.edu/pub/ietf/http/related/iso639.txt
#
http://userpage.chemie.fu-berlin.de/diverse/doc/ISO_3166.html
#
locales=ar_SA,ca_AD,zh_CN,zh_TW,cs_CZ,nl_NL,en_US,fi_FI,fr_FR,de_DE,el_GR,hu_HU,it
#
# Set the following to true if unauthenticated users get their
preferred
# language from the Accept-Language header. Set the following to
false if
# unauthenticated users get their preferred language from their
company.
#
locale.default.request=false
#
# Specify the Struts character encoding. UTF-8 allows for the use
of more
# languages but takes a 15% performance hit compared to
ISO-8859-1.
#
struts.char.encoding=UTF-8
#
# Specify the available time zones. The specified ids must match
those from
# the class java.util.TimeZone.
#
time.zones=Pacific/Midway,Pacific/Honolulu,AST,PST,MST,CST,EST,PRT,CNT,BET,America
GMT,WET,CET,EET,Asia/Jerusalem,Asia/Baghdad,Iran,Asia/Dubai,Asia/Kabul,Asia/Karach
Asia/Dhaka,Asia/Rangoon,VST,CTT,JST,ROK,ACT,AET,SST,NST,Pacific/Enderbury,Pacific/
##
## Look and Feel
##
#
# Set the following to false if the system does not use allow
users to
# modify the look and feel.
#
look.and.feel.modifiable=true
#
# Set the default theme id.
#
default.theme.id=classic
#
# Set the default color scheme id.
#
default.color.scheme.id=01
##
## Request
##
#
# Portlets that have been configured to use private request
attributes in
```



```
# liferay-portlet.xml may still want to share some request
attributes. This
# property allows you to configure which request attributes will
be shared.
# Set a comma delimited list of attribute names that will be
shared when the
# attribute name starts with one of the specified attribute names.
For
# example, if you set the value to "hello_,world_", then all
attribute names
# that start with "hello_" or "world_" will be shared.
#
request.shared.attributes=LIFERAY_SHARED_

##
## Session
##

#
# Specify the number of minutes before a session expires. This
value is
# always overridden by the value set in web.xml.
#
session.timeout=30

#
# Specify the number of minutes before a warning is sent to the
user
# informing the user of the session expiration. Specify 0 to
disable any
# warnings.
#
session.timeout.warning=1

#
# Portlets that have been configured to use private session
attributes in
# liferay-portlet.xml may still want to share some session
attributes. This
# property allows you to configure which session attributes will
be shared.
# Set a comma delimited list of attribute names that will be
shared when the
# attribute name starts with one of the specified attribute names.
For
# example, if you set the value to "hello_,world_", then all
attribute names
# that start with "hello_" or "world_" will be shared.
#
# Note that this property is used to specify the sharing of
session
# attributes from the portal to the portlet. This is not used to
specify
# session sharing between portlet WARs or from the portlet to the
portal.
#
session.shared.attributes=org.apache.struts.action.LOCALE,COMPANY_,USER_

#
# Set this to false to disable all persistent cookie. Features
like
# automatically logging in will not work.
#
session.enable.persistent.cookies=true
```

```
#
# Input a list of comma delimited class names that extend
# com.liferay.portal.struts.SessionAction. These classes will run
at the
# specified event.
#
#
# Servlet session create event
#
servlet.session.create.events=com.liferay.portal.events.SessionCreateAction

#
# Servlet session destroy event
#
servlet.session.destroy.events=com.liferay.portal.events.SessionDestroyAction

#
# Set the following to true to track user clicks in memory for the
duration
# of a user's session. Setting this to true allows you to view all
live
# sessions in the Admin portlet.
#
session.tracker.memory.enabled=true

#
# Set the following to true to track user clicks in the database
after a
# user's session is invalidated. Setting this to true allows you
to generate
# usage reports from the database. Use this cautiously because
this will
# store a lot of usage data.
#
session.tracker.persistence.enabled=false

##
## JAAS
##

#
# Specify a class name that implements
# com.liferay.portal.security.auth.PrincipalFinder. A custom
version of this
# class is needed for portals that integrate with external EJBs
that have a
# different scheme for storing principals. This will allow the
portal to
# dynamically convert the principal of an external system into the
unique
# user id of the portal system.
#
principal.finder=com.liferay.portal.security.auth.BasicPrincipalFinder

#
# Set the following to true if the portal will use
# com.liferay.portal.security.jaas.PortalConfiguration as the JAAS
master
# configuration.
#
portal.configuration=true
```

```
#
# Set the following to false to disable JAAS security checks.
Disabling JAAS
# speeds up login. JAAS must be disabled if administrators are to
be able to
# impersonate other users.
#
portal.jaas.enable=false

#
# By default, com.liferay.portal.security.jaas.PortalLoginModule
loads the
# correct JAAS login module based on what application server or
servlet
# container the portal is deployed on. Set a JAAS implementation
class to
# override this behavior.
#
#portal.jaas.impl=

#
# Set the following to true to enable administrators to
impersonate other
# users. JAAS must also be disabled for this feature to work.
#
portal.impersonation.enable=true

##
## LDAP Import
##

ldap.import.enabled=false
ldap.import.on.startup=false

#
# Enter time in minutes. This is how often the importer will
synchronize
# with LDAP. This property is portal wide. Company override will
be ignored.
#
ldap.import.interval=10

ldap.import.factory.initial=com.sun.jndi.ldap.LdapCtxFactory
ldap.import.base.provider.url=ldap://localhost:10389
ldap.import.base.dn=dc=example,dc=com
ldap.import.security.principal=uid=admin,ou=system
ldap.import.security.credentials=secret
ldap.import.search.filter=(objectClass=inetOrgPerson)
ldap.import.user.mappings=userId=cn\npassword=userPassword\nemailAddress=mail\nfir
lastName=sn\njobTitle=title\ngroup=groupMembership
ldap.import.group.mappings=groupName=cn\ndescription=description

##
## Authentication Pipeline
##

#
# Input a list of comma delimited class names that implement
# com.liferay.portal.security.auth.Authenticator. These classes
will run
# before or after the portal authentication begins.
#
# The Authenticator class defines the constant values that should
be used
```

```
# as return codes from the classes implementing the interface. If
# authentication is successful, return SUCCESS; if the user exists
but the
# passwords do not match, return FAILURE; and if the user does not
exist on
# the system, return DNE.
#
# Constants in Authenticator:
#     public static final int SUCCESS = 1;
#     public static final int FAILURE = -1;
#     public static final int DNE = 0;
#
# In case you have several classes in the authentication pipeline,
all of
# them have to return SUCCESS if you want the user to be able to
login. If
# one of the authenticators returns FAILURE or DNE, the login
fails.
#
# Under certain circumstances, you might want to keep the
information in the
# portal database in sync with an external database or an LDAP
server. This
# can easily be achieved by implementing a class via LDAPAuth that
updates
# the information stored in the portal user database whenever a
user signs
# in.
#
# Each portal instance can be configured at run time to either
authenticate
# based on user ids or email addresses. See the Admin portlet for
more
# information.
#
# Available authenticators are:
#     com.liferay.portal.security.auth.LDAPAuth
#
auth.pipeline.pre=com.liferay.portal.security.auth.LDAPAuth
#auth.pipeline.post=

#
# Set this to true to enable password checking by the internal
portal
# authentication. If set to false, you're essentially delegating
password
# checking is delegated to the authenticators configured in
# "auth.pipeline.pre" and "auth.pipeline.post" settings.
#
auth.pipeline.enable.liferay.check=true

#
# LDAPAuth can be configured from the Admin portlet. It provides
out of the
# box support for Apache Directory Server, Microsoft Active
Directory
# Server, Novell eDirectory, and OpenLDAP. The default settings
are for
# Apache Directory Server.
#
# Encryption is implemented by
com.liferay.util.Encryptor.provider.class in
# system.properties.
#
```

```

auth.impl.ldap.enabled=false
auth.impl.ldap.required=false
auth.impl.ldap.factory.initial=${ldap.import.factory.initial}
auth.impl.ldap.base.provider.url=${ldap.import.base.provider.url}
#auth.impl.ldap.base.dn=${ldap.import.base.dn}
auth.impl.ldap.base.dn=dc=example,dc=com
#auth.impl.ldap.security.principal=${ldap.import.security.principal}
auth.impl.ldap.security.principal=uid=admin,ou=system
auth.impl.ldap.security.credentials=${ldap.import.security.credentials}
auth.impl.ldap.search.filter=(mail=@email_address@)
auth.impl.ldap.password.encryption.algorithm=
auth.impl.ldap.password.encryption.algorithm.types=MD5,SHA
auth.impl.ldap.user.mappings=${ldap.import.user.mappings}

#
# Input a list of comma delimited class names that implement
# com.liferay.portal.security.auth.AuthFailure. These classes will
run when
# a user has a failed login or when a user has reached the maximum
number of
# failed logins.
#
auth.failure=com.liferay.portal.security.auth.LoginFailure
auth.max.failures=com.liferay.portal.security.auth.LoginMaxFailures
auth.max.failures.limit=5

#
# Set the following to true if users are allowed to have
simultaneous logins
# from different sessions.
#
auth.simultaneous.logins=true

#
# Set the following to true if users are forwarded to the last
visited path
# upon successful login. If set to false, users will be forwarded
to their
# default layout page.
#
auth.forward.by.last.path=false

#
# Enter a list of paths that can be considered part of the last
visited
# path.
#
auth.forward.last.path.0=/document_library/get_file

#
# Enter a list of paths that do not require authentication.
#
auth.public.path.0=/blogs/find_entry
auth.public.path.1=/blogs/rss
auth.public.path.2=/document_library/get_file
auth.public.path.3=/google_maps/save_destination_address
auth.public.path.4=/google_maps/save_source_address
auth.public.path.5=/journal/get_articles
auth.public.path.6=/journal/get_latest_article_content
auth.public.path.7=/journal/get_structure
auth.public.path.8=/journal/get_template
auth.public.path.9=/journal_articles/view_article_content
auth.public.path.9=/message_boards/find_category
auth.public.path.10=/message_boards/find_message

```

```
auth.public.path.11=/message_boards/find_thread
auth.public.path.12=/message_boards/get_message_attachment
auth.public.path.13=/message_boards/rss
auth.public.path.14=/messaging/action
auth.public.path.15=/polls/view_chart
auth.public.path.16=/portal/json_regions
auth.public.path.17=/portal/session_click
auth.public.path.18=/portal/session_tree_js_click
auth.public.path.19=/shopping/notify

##
## Auto Login
##

#
# Input a list of comma delimited class names that implement
# com.liferay.portal.security.auth.AutoLogin. These classes will
run in
# consecutive order for all unauthenticated users until one of
them return a
# valid user id and password combination. If no valid combination
is
# returned, then the request continues to process normally. If a
valid
# combination is returned, then the portal will automatically
login that
# user with the returned user id and password combination.
#
# For example, com.liferay.portal.security.auth.BasicAutoLogin
reads from a
# cookie to automatically log in a user who previously logged in
while
# checking on the "Remember Me" box.
#
# This interface allows deployers to easily configure the portal
to work
# with other SSO servers. See
com.liferay.portal.security.auth.CASAutoLogin
# for an example of how to configure the portal with Yale's SSO
server.
#
auto.login.hooks=com.liferay.portal.security.auth.BasicAutoLogin
#auto.login.hooks=com.liferay.portal.security.auth.BasicAutoLogin,com.liferay.port

##
## SSO with MAC (Message Authentication Code)
##

#
# To use SSO with MAC, post to an URL like:
#
http://localhost:8080/c/portal/login?cmd=already-registered&login=<userId|emailAdd
#
# Pass the MAC in the password field. Make sure the MAC gets URL
encoded
# because it might contain characters not allowed in a URL.
#
# SSO with MAC also requires that you set the following property
in
# system.properties:
#
#     com.liferay.util.servlet.SessionParameters=false
#
# See the following links:
```

```
#      http://support.liferay.com/browse/LEP-1288
#      http://en.wikipedia.org/wiki/Message_authentication_code
#
# Set the following to true to enable SSO with MAC.
#
auth.mac.allow=false

#
# Set the algorithm to use for MAC encryption.
#
auth.mac.algorithm=MD5

#
# Set the shared key used to generate the MAC.
#
auth.mac.shared.key=

##
## Passwords
##

#
# Set the following to true to encrypt passwords using a MD5 hash
algorithm.
# If set to false, passwords are stored in the database as plain
text.
#
passwords.encrypted=true

#
# Input a class name that extends
# com.liferay.portal.security.pwd.BasicToolkit. This class will be
called to
# generate and validate passwords.
#
passwords.toolkit=com.liferay.portal.security.pwd.RegExpToolkit

#
# If you choose to use the default
# com.liferay.portal.security.pwd.RegExpToolkit as your passwords
toolkit,
# set the regular expression pattern that will be used to generate
and
# validate passwords.
#
# Note that \ is replaced with \\ to work in Java.
#
# The first pattern ensures that passwords must have at least 4
valid
# characters consisting of digits or letters.
#
# The second pattern ensures that passwords must have at least 8
valid
# characters consisting of digits or letters.
#
passwords.regexptoolkit.pattern=(?={4})(?:[a-zA-Z0-9]*)
#passwords.regexptoolkit.pattern=(?={8})(?:[a-zA-Z0-9]*)

#
# Set the following to true if passwords can be a dictionary word.
#
passwords.allow.dictionary.word=true

#
```

```
# Set the following to true if users ought to change their
passwords on
# first use when an Administrator creates their account.
#
passwords.change.on.first.use=false

#
# Set the number of days that will pass before users are prompted
to change
# their password. Set the number of days to 0 if passwords never
expire.
#
passwords.lifespan=0

#
# Set the number of days that must have passed before a password
is allowed
# to be recycled and used again. Set the number of days to 0 if
passwords
# can always be reused.
#
passwords.recycle=0

##
## Permissions
##

#
# Set the default permission checker class used by
# com.liferay.portal.security.permission.PermissionCheckerFactory
to check
# permissions for actions on objects. These classes can be
overridden with
# custom classes that extend
# com.liferay.portal.security.permission.PermissionCheckerImpl.
#
permissions.checker=com.liferay.portal.security.permission.PermissionCheckerImpl

#
# Set the algorithm used to check permissions for a user. This is
useful so
# that you can optimize the search for different databases. See
# com.liferay.portal.service.impl.PermissionLocalServiceImpl.
#
#permissions.user.check.algorithm=1
permissions.user.check.algorithm=2
#permissions.user.check.algorithm=3
#permissions.user.check.algorithm=4

##
## Captcha
##

#
# Set the maximum number of captcha checks per portlet session.
Set this
# value to 0 to always check. Set this value to a number less than
0 to
# never check.
#
captcha.max.challenges=1

##
## Startup Events
```



```
##
#
# Input a list of comma delimited class names that extend
# com.liferay.portal.struts.SimpleAction. These classes will run
at the
# specified event.
#
#
# Global startup event that runs once when the portal initializes.
#
global.startup.events=com.liferay.portal.events.GlobalStartupAction
#
# Application startup event that runs once for every web site
instance of
# the portal that initializes.
#
application.startup.events=com.liferay.portal.events.AppStartupAction
##
## Shutdown Events
##
#
# Input a list of comma delimited class names that extend
# com.liferay.portal.struts.SimpleAction. These classes will run
at the
# specified event.
#
#
# Global shutdown event that runs once when the portal shuts down.
#
global.shutdown.events=com.liferay.portal.events.GlobalShutdownAction
#
# Application shutdown event that runs once for every web site
instance of
# the portal that shuts down.
#
application.shutdown.events=com.liferay.portal.events.AppShutdownAction
##
## Portal Events
##
#
# Input a list of comma delimited class names that extend
# com.liferay.portal.struts.Action. These classes will run before
or after
# the specified event.
#
#
# Servlet service event (The pre-service events have an associated
error
# page and will forward to that page if an exception is thrown
during
# execution of the events. The pre-service events process before
Struts
# processes the request. The post-service events process after
Struts
```

```

    # processes the request.
    #
servlet.service.events.pre=com.liferay.portal.events.ServicePreAction
#servlet.service.events.pre=com.liferay.portal.events.LogMemoryUsageAction,com.lif
com.liferay.portal.events.ServicePreAction
#servlet.service.events.pre=com.liferay.portal.events.LogSessionIdAction,com.lifer
#servlet.service.events.pre=com.liferay.portal.events.ServicePreAction,com.liferay
#servlet.service.events.pre=com.liferay.portal.events.ServicePreAction,com.liferay
    #servlet.service.events.pre.error.page=/common/error.jsp
servlet.service.events.post=com.liferay.portal.events.ServicePostAction

#
# Login event
#
login.events.pre=com.liferay.portal.events.LoginPreAction
login.events.post=com.liferay.portal.events.LoginPostAction

#
# Logout event
#
logout.events.pre=com.liferay.portal.events.LogoutPreAction
logout.events.post=com.liferay.portal.events.LogoutPostAction,com.liferay.portal.e

##
## Default Guest
##

#
# The Guest group at least one public page. The settings for the
initial
# public page are specified in the following properties. For more
complex
# behavior, override the addDefaultLayouts method in
# com.liferay.portal.service.impl.GroupLocalServiceImpl.
#

#
# Set the layout name.
#
default.guest.layout.name=Welcome

#
# Set the layout template id that matches an existing TPL.
#
#default.guest.layout.template.id=1_2_1_columns
#default.guest.layout.template.id=1_column
#default.guest.layout.template.id=2_2_columns
#default.guest.layout.template.id=2_columns_i
default.guest.layout.template.id=2_columns_ii
#default.guest.layout.template.id=2_columns_iii
#default.guest.layout.template.id=3_columns

#
# Set the layout ids for the column specified in the layout
template.
#
default.guest.layout.column-1=58,
default.guest.layout.column-2=47,
default.guest.layout.column-3=
default.guest.layout.column-4=

#
# Enter either "800x600" or "1024x768" to set the layout
resolution.

```

```
#
default.guest.layout.resolution=800x600

##
## Default User
##

#
# Users who have the Power User role must have at least one
private personal
# page. The settings for the initial private page are specified in
the
# following properties. For more complex behavior, override the
# addDefaultLayouts method in
com.liferay.portal.events.ServicePreAction.
#

#
# Set the layout name.
#
default.user.layout.name=Home

#
# Set the layout template id that matches an existing TPL.
#
#default.user.layout.template.id=1_2_1_columns
#default.user.layout.template.id=1_column
#default.user.layout.template.id=2_2_columns
#default.user.layout.template.id=2_columns_i
default.user.layout.template.id=2_columns_ii
#default.user.layout.template.id=2_columns_iii
#default.user.layout.template.id=3_columns

#
# Set the layout ids for the column specified in the layout
template.
#
default.user.layout.column-1=82,23,61,65,
default.user.layout.column-2=8,11,36,33,
default.user.layout.column-3=
default.user.layout.column-4=

#
# Enter either "800x600" or "1024x768" to set the layout
resolution.
#
default.user.layout.resolution=800x600

##
## Default Admin
##

#
# Set the default admin password.
#
default.admin.password=test

#
# Set the default admin email address prefix.
#
default.admin.email.address.prefix=test

#
# Set the default admin first name.
```

```

#
default.admin.first.name=Test

#
# Set the default admin middle name.
#
default.admin.middle.name=

#
# Set the default admin last name.
#
default.admin.last.name=Test

##
## Layouts
##

#
# Set the list of layout types. The display text of each of the
layout types
# is set in content/Language.properties and prefixed with
"layout.types.".
#
# You can create new layout types and specify custom settings for
each
# layout type. End users input dynamic values as designed in the
edit page.
# End users see the layout as designed in the view page. The
generated
# URL can reference properties set in the edit page. Parentable
layouts
# can contain child layouts.
#
layout.types=portlet,embedded,article,url

#
# Default settings layouts.
#
layout.edit.page=/portal/layout/edit/portlet.jsp
layout.view.page=/portal/layout/view/portlet.jsp
layout.url=${liferay:mainPath}/portal/layout?p_l_id=${liferay:plid}
layout.url.friendliable=true
layout.parentable=true

#
# Settings for portlet layouts are inherited from the default
settings.
#
#layout.edit.page[portlet]=/portal/layout/edit/portlet.jsp
#layout.view.page[portlet]=/portal/layout/view/portlet.jsp
#layout.url[portlet]=${liferay:mainPath}/portal/layout?p_l_id=${liferay:plid}
#layout.url.friendliable[portlet]=true
#layout.parentable[portlet]=true

#
# Settings for embedded layouts.
#
layout.edit.page[embedded]=/portal/layout/edit/embedded.jsp
layout.view.page[embedded]=/portal/layout/view/embedded.jsp
layout.url[embedded]=${liferay:mainPath}/portal/layout?p_l_id=${liferay:plid}
layout.url.friendliable[embedded]=true
layout.parentable[embedded]=false

#

```

```

# Settings for article layouts.
#
layout.edit.page[article]=/portal/layout/edit/article.jsp
layout.view.page[article]=/portal/layout/view/article.jsp
layout.url.friendly[article]=true
layout.url[article]=${liferay:mainPath}/portal/layout?p_l_id=${liferay:plid}
layout.parentable[article]=false

#
# Settings for URL layouts.
#
layout.edit.page[url]=/portal/layout/edit/url.jsp
layout.view.page[url]=
layout.url[url]=${url}
layout.url.friendly[url]=true
layout.parentable[url]=false

#
# Specify static portlets that cannot be moved and will always
appear on
# every layout. Static portlets will take precedence over portlets
that may
# have been dynamically configured for the layout.
#
# For example, if you want the Hello World portlet to always
appear at the
# start of the iteration of the first column for user layouts, set
the
# property "layout.static.portlets.start.column-1[user]" to "47".
If you
# want the Hello World portlet to always appear at the end of the
second
# column for user layouts, set the property
# "layout.static.portlets.end.column-2[user]" to "47". You can
input a
# list of comma delimited portlet ids to specify more than one
portlet.
#
# The static portlets are fetched based on the properties
controlled by
# custom filters using EasyConf. By default, the available filters
are
# user, community, and organization.
#
#layout.static.portlets.start.column-1[user]=3,6
#layout.static.portlets.end.column-1[user]=14
#layout.static.portlets.start.column-2[user]=
#layout.static.portlets.end.column-2[user]=
#layout.static.portlets.start.column-3[user]=7
#layout.static.portlets.end.column-3[user]=8

#
# It is also possible to add a static portlet which only shows in
the first
# layout of a user or community.
#
#layout.static.portlets.start.column-1[user][firstLayout]=3,6
#layout.static.portlets.end.column-2[community][firstLayout]=14

#
# Set the static layouts for community layouts.
#
#layout.static.portlets.start.column-1[community]=
#layout.static.portlets.end.column-1[community]=

```

```

#layout.static.portlets.start.column-2[community]=
#layout.static.portlets.end.column-2[community]=
#layout.static.portlets.start.column-3[community]=
#layout.static.portlets.end.column-3[community]=

#
# Set the static layouts for organization layouts.
#
#layout.static.portlets.start.column-1[organization]=
#layout.static.portlets.end.column-1[organization]=
#layout.static.portlets.start.column-2[organization]=
#layout.static.portlets.end.column-2[organization]=
#layout.static.portlets.start.column-3[organization]=
#layout.static.portlets.end.column-3[organization]=

#
# Set the public and private servlet mapping for
# com.liferay.portal.servlet.FriendlyURLServlet. This value must
match the
# servlet mapping set in web.xml.
#
# For example, if the private pages are mapped to "/group" and the
group's
# friendly URL is set to "/guest" and the layout's friendly URL is
set to
# "/company/community", then the friendly URL for the page will be
# http://www.liferay.com/group/guest/company/community. Private
pages are
# only available to authenticated users with the proper
permissions.
#
# For example, if the public pages are mapped to "/web" and the
group's
# friendly URL is set to "/guest" and the layout's friendly URL is
set to
# "/company/community", then the friendly URL for the page will be
# http://www.liferay.com/web/guest/company/community. Public pages
are
# available to unauthenticated users.
#
# The friendly URL's for users, groups, and layouts can be set
during
# runtime.
#
layout.friendly.url.private.servlet.mapping=/group
layout.friendly.url.public.servlet.mapping=/web

#
# Redirect to this resource if the user requested a friendly URL
that does
# not exist. Leave it blank to display nothing.
#
#layout.friendly.url.page.not.found=/html/portal/404.html

#
# Set the reserved keywords that cannot be used in a friendly URL.
#
layout.friendly.url.keywords=c,group,web,image,wsrp,page,public,private,blogs,calendar,
image_gallery,journal,message_boards,podcast,rss,wiki

#
# Set the following to true if users are allowed to add portlets
from the
# layout page.

```

```
#
layout.add.portlets=true

#
# Set the maximum length to display a layout name.
#
layout.name.max.length=10

#
# Set the maximum number of tabs per row.
#
layout.tabs.per.row=7

#
# Set the following to true if layouts should remember (across
sessions)
# that a window state was set to maximized.
#
layout.remember.session.window.state.maximized=false

#
# Set the following to true if layouts should remember (across
requests)
# that a window state was set to maximized.
#
layout.remember.request.window.state.maximized=true

#
# Set the following to true if guest users should see the maximize
window
# icon.
#
layout.guest.show.max.icon=false

#
# Set the following to true if guest users should see the minimize
window
# icon.
#
layout.guest.show.min.icon=false

#
# Set the following to true if users are shown that they do not
have access
# to a portlet. The portlet init parameter
"show-portlet-access-denied" will
# override this setting.
#
layout.show.portlet.access.denied=true

#
# Set the following to true if users are shown that a portlet is
inactive.
# The portlet init parameter "show-portlet-inactive" will override
this
# setting.
#
layout.show.portlet.inactive=true

#
# Set the default layout template id used when creating layouts.
#
layout.default.template.id=2_columns_ii
```

```
#
# Set the following to false to disable parallel rendering. You
can also
# disable it on a per request basis by setting the attribute key
# com.liferay.portal.util.WebKeys.PORTLET_PARALLEL_RENDER to the
# Boolean.FALSE in a pre service event.
#
layout.parallel.render.enable=true

##
## Portlet URL
##

#
# Set the following to true if calling setParameter on a portlet
URL appends
# the parameter value versus replacing it. There is some
disagreement in the
# interpretation of the JSR 168 spec among portlet developers over
this
# specific behavior. Liferay Portal successfully passes the
portlet TCK
# tests whether this value is set to true or false.
#
# See http://support.liferay.com/browse/LEP-426 for more
information.
#
portlet.url.append.parameters=false

#
# Set the following to true to allow portlet URLs to generate with
an anchor
# tag.
#
portlet.url.anchor.enable=false

##
## Preferences
##

#
# Set the following to true to validate portlet preferences on
startup.
#
preference.validate.on.startup=false

##
## Struts
##

#
# Input the custom Struts request processor that will be used by
Struts
# based portlets. The custom class must extend
# com.liferay.portal.struts.PortletRequestProcessor and have the
same
# constructor.
#
struts.portlet.request.processor=com.liferay.portal.struts.PortletRequestProcessor

##
## Images
##
```



```
#
# Set the location of the default spacer image that is used for
missing
# images. This image must be found in the class path.
#
image.default.spacer=com/liferay/portal/dependencies/spacer.gif

#
# Set the location of the default user portrait image that is used
for
# missing user portrait images. This image must be found in the
class path.
#
image.default.user.portrait=com/liferay/portal/dependencies/user_portrait.gif

##
## Editors
##

#
# You can configure individual JSP pages to use a specific
implementation of
# the available WYSIWYG editors: liferay, fckeditor, simple,
tinymce,
# or tinymce-simple.
#
editor.wysiwyg.default=fckeditor
editor.wysiwyg.portal-web.docroot.html.portlet.admin.users.jsp=fckeditor
editor.wysiwyg.portal-web.docroot.html.portlet.blogs.edit_entry.jsp=fckeditor
editor.wysiwyg.portal-web.docroot.html.portlet.calendar.edit_configuration.jsp=fck
editor.wysiwyg.portal-web.docroot.html.portlet.journal.edit_article_content.jsp=fc
editor.wysiwyg.portal-web.docroot.html.portlet.journal.edit_article_content_xsd_el
editor.wysiwyg.portal-web.docroot.html.portlet.journal.edit_configuration.jsp=fcke
editor.wysiwyg.portal-web.docroot.html.portlet.mail.edit.jsp=fckeditor
editor.wysiwyg.portal-web.docroot.html.portlet.mail.edit_message.jsp=fckeditor
editor.wysiwyg.portal-web.docroot.html.portlet.message_boards.edit_configuration.j
editor.wysiwyg.portal-web.docroot.html.portlet.shopping.edit_configuration.jsp=fck
editor.wysiwyg.portal-web.docroot.html.portlet.wiki.edit_page.jsp=fckeditor

##
## Fields
##

#
# Set the following fields to false so users cannot see them. Some
company
# policies require gender and birthday information to always be
hidden.
#
field.enable.com.liferay.portal.model.Contact.male=true
field.enable.com.liferay.portal.model.Contact.birthday=true

##
## Amazon License Keys
##

#
# Enter a list of valid Amazon license keys. Configure additional
keys by
# incrementing the last number. The keys are used following a
Round-Robin
# algorithm. This is made available only for personal use. Please
see the
# Amazons license at http://www.amazon.com for more information.
```

```
#
#amazon.license.0=
#amazon.license.1=
#amazon.license.2=
#amazon.license.3=

##
## Google License Keys
##

#
# Enter a list of valid Google license keys. Configure additional
keys by
# incrementing the last number. Each key is valid for 1000
requests. This is
# made available only for personal use. Please see the Google
license at
# http://www.google.com/apis for more information.
#
#google.license.0=
#google.license.1=
#google.license.2=
#google.license.3=

#
# Enter a valid Google Maps license key for your domain. See
# http://www.google.com/apis/maps for more information.
#
#google.maps.license=

##
## Instant Messenger
##

#
# Set the AIM login and password by which the system will use AIM
to
# communicate with users.
#
aim.login=
aim.password=

#
# Due to a bug in JOscarLib 0.3b1, you must set the full path to
the ICQ
# jar.
#
# See the following posts:
#   http://sourceforge.net/forum/message.php?msg\_id=1972697
#   http://sourceforge.net/forum/message.php?msg\_id=1990487
#
icq.jar=C:/Java/orion-2.0.7/lib/icq.jar

#
# Set the ICQ login and password by which the system will use ICQ
to
# communicate with users.
#
icq.login=
icq.password=

#
# Set the MSN login and password by which the system will use MSN
to
```

```

# communicate with users.
#
msn.login=
msn.password=

#
# Set the YM login and password by which the system will use YM to
# communicate with users.
#
ym.login=
ym.password=

##
## Lucene Search
##

#
# Set the following to true if you want to index your entire
library of
# files on startup.
#
index.on.startup=true

#
# Set the following to true if you want the indexing on startup to
be
# executed on a separate thread to speed up execution.
#
index.with.thread=false

#
# Designate whether Lucene stores indexes in the file system or in
the
# database.
#
#lucene.store.type=jdbc
lucene.store.type=file

#
# Set the JDBC dialect that Lucene uses to store indexes in the
database.
# This is only referenced if Lucene stores indexes in the
database. Liferay
# will attempt to load the proper dialect based on the URL of the
JDBC
# connection. For example, the property
"lucene.store.jdbc.dialect.mysql" is
# read for the JDBC connection URL
"jdbc:mysql://localhost/lportal".
#
lucene.store.jdbc.dialect.db2=org.apache.lucene.store.jdbc.dialect.DB2Dialect
lucene.store.jdbc.dialect.hsqldb=org.apache.lucene.store.jdbc.dialect.HSQLDialect
lucene.store.jdbc.dialect.jtds=org.apache.lucene.store.jdbc.dialect.SQLServerDialect
lucene.store.jdbc.dialect.microsoft=org.apache.lucene.store.jdbc.dialect.SQLServer
lucene.store.jdbc.dialect.mysql=org.apache.lucene.store.jdbc.dialect.MySQLDialect
lucene.store.jdbc.dialect.oracle=org.apache.lucene.store.jdbc.dialect.OracleDialect
lucene.store.jdbc.dialect.postgresql=org.apache.lucene.store.jdbc.dialect.PostgreSQL

#
# Set the directory where Lucene indexes are stored. This is only
referenced
# if Lucene stores indexes in the file system.
#
lucene.dir=/home/liferay/lucene/

```

```

#
# Input a class name that extends
# com.liferay.portal.lucene.LuceneFileExtractor. This class is
called by
# Lucene to extract text from complex files so that they can be
properly
# indexed.
#
lucene.file.extractor=com.liferay.portal.lucene.LuceneFileExtractor

#
# Set the default analyzer used for indexing and retrieval.
#
#lucene.analyzer=org.apache.lucene.analysis.br.BrazilianAnalyzer
#lucene.analyzer=org.apache.lucene.analysis.cn.ChineseAnalyzer
#lucene.analyzer=org.apache.lucene.analysis.cjk.CJKAnalyzer
#lucene.analyzer=org.apache.lucene.analysis.cz.CzechAnalyzer
#lucene.analyzer=org.apache.lucene.analysis.nl.DutchAnalyzer
#lucene.analyzer=org.apache.lucene.analysis.fr.FrenchAnalyzer
#lucene.analyzer=org.apache.lucene.analysis.de.GermanAnalyzer
#lucene.analyzer=org.apache.lucene.analysis.KeywordAnalyzer
#lucene.analyzer=org.apache.lucene.index.memory.PatternAnalyzer
#lucene.analyzer=org.apache.lucene.analysis.PerFieldAnalyzerWrapper
#lucene.analyzer=org.apache.lucene.analysis.ru.RussianAnalyzer
#lucene.analyzer=org.apache.lucene.analysis.SimpleAnalyzer
#lucene.analyzer=org.apache.lucene.analysis.snowball.SnowballAnalyzer
lucene.analyzer=org.apache.lucene.analysis.standard.StandardAnalyzer
#lucene.analyzer=org.apache.lucene.analysis.StopAnalyzer
#lucene.analyzer=org.apache.lucene.analysis.WhitespaceAnalyzer

##
## Value Object
##

#
# You can add a listener for a specific class by setting the
property
# "value.object.listener" plus the class name to a class that
implements
# com.liferay.portal.model.ModelListener.
#
value.object.listener.com.liferay.portal.model.Layout=com.liferay.portal.model.Lay
value.object.listener.com.liferay.portal.model.LayoutSet=com.liferay.portal.model.
value.object.listener.com.liferay.portal.model.PortletPreferences=com.liferay.port
value.object.listener.com.liferay.portal.model.User=com.liferay.portal.model.UserL
value.object.listener.com.liferay.portlet.journal.model.JournalArticle=com.liferay
value.object.listener.com.liferay.portlet.journal.model.JournalTemplate=com.lifera

##
## Last Modified
##

#
# Set the following to true to check last modified date on server
side CSS
# and JavaScript.
#
last.modified.check=true

#
# Enter a list of paths that will only be executed when newer than
the last
# modified date. These paths must extend

```

```

# com.liferay.portal.lastmodified.LastModifiedAction.
#
last.modified.path.0=/portal/css_cached
last.modified.path.1=/portal/javascript_cached

##
## XSS (Cross Site Scripting)
##

#
# Set the following to false to ensure that all persisted data is
stripped
# of XSS hacks.
#
xss.allow=false

#
# You can override the "xss.allow" setting for a specific class by
setting
# the property "xss.allow" plus the class name.
#
xss.allow.com.liferay.portal.model.Portlet=true
xss.allow.com.liferay.portal.model.PortletPreferences=true

#
# You can override the "xss.allow" setting for a specific field in
a class
# by setting the property "xss.allow" plus the class and field
name.
#
xss.allow.com.liferay.portlet.journal.model.JournalArticle.content=true
xss.allow.com.liferay.portlet.journal.model.JournalStructure.xsd=true
xss.allow.com.liferay.portlet.journal.model.JournalTemplate.xsl=true

##
## Counter Server
##

#
# Uncomment the following and set the appropriate values if the
server
# resides on a remote server. Make sure the EJBs are available on
the remote
# server and not on the portal server.
#
#com_liferay_counter_ejb_CounterManager_java.naming.factory.initial=com.evermind.s
#com_liferay_counter_ejb_CounterManager_java.naming.provider.url=ormi://host/default
#com_liferay_counter_ejb_CounterManager_java.naming.security.principal=admin
#com_liferay_counter_ejb_CounterManager_java.naming.security.credentials=1234

##
## Lock Server
##

#
# Uncomment the following and set the appropriate values if the
server
# resides on a remote server. Make sure the EJBs are available on
the remote
# server and not on the portal server.
#
#com_liferay_lock_ejb_LockManager_java.naming.factory.initial=com.evermind.server.
#com_liferay_lock_ejb_LockManager_java.naming.provider.url=ormi://host/default
#com_liferay_lock_ejb_LockManager_java.naming.security.principal=admin

```

```
#com_liferay_lock_ejb_LockManager_java.naming.security.credentials=1234

#
# Set the lock expiration time for each class.
#

#
# 1 day
#
lock.expiration.time.com.liferay.portlet.documentlibrary.model.DLFileEntryModel=86400

#
# 20 minutes
#
lock.expiration.time.com.liferay.portlet.wiki.model.WikiPageModel=120000

##
## Jabber
##

jabber.xmlpp.server.enabled=false
jabber.xmlpp.server.address=localhost
jabber.xmlpp.server.name=localhost
jabber.xmlpp.server.port=5222
jabber.xmlpp.user.password=L1f3RayJabb3r

##
## JBI
##

jbi.workflow.url=http://localhost:8080/servicemix-web/workflow

##
## JCR
##

jcr.initialize.on.startup=false

jcr.workspace.name=liferay
jcr.node.documentlibrary=documentlibrary

jcr.jackrabbit.repository.root=/home/liferay/jackrabbit
jcr.jackrabbit.config.file.path=${jcr.jackrabbit.repository.root}/repository.xml
jcr.jackrabbit.repository.home=${jcr.jackrabbit.repository.root}/home
jcr.jackrabbit.credentials.username=none
jcr.jackrabbit.credentials.password=none

##
## Reverse AJAX
##

reverse.ajax.enabled=false
reverse.ajax.heartbeat=180000

##
## SMTP
##

smtp.server.enabled=true
smtp.server.port=48625
smtp.server.subdomain=events

##
## Web Server
```

```
##
#
# Set the HTTP and HTTPS ports when running the portal in a J2EE
server that
# is sitting behind another web server like Apache. Set the values
to -1 if
# the portal is not running behind another web server like Apache.
#
web.server.http.port=-1
web.server.https.port=-1

#
# Set the hostname that will be used when the portlet generates
URLs.
# Leaving this blank will mean the host is derived from the
servlet
# container.
#
web.server.host=

#
# Set the preferred protocol.
#
#web.server.protocol=https

##
## Main Servlet
##

#
# Servlets can be protected by
com.liferay.filters.secure.SecureFilter.
#
# Input a list of comma delimited IPs that can access this
servlet. Input a
# blank list to allow any IP to access this servlet. SERVER_IP
will be
# replaced with the IP of the host server.
#
main.servlet.hosts.allowed=

#
# Set the following to true if this servlet can only be accessed
via https.
#
main.servlet.https.required=false

##
## Axis Servlet
##

#
# See Main Servlet on how to protect this servlet.
#
axis.servlet.hosts.allowed=127.0.0.1,SERVER_IP
axis.servlet.https.required=false

##
## Liferay Tunnel Servlet
##

#
# See Main Servlet on how to protect this servlet.
```

```

#
tunnel.servlet.hosts.allowed=127.0.0.1,SERVER_IP
tunnel.servlet.https.required=false

##
## Spring Remoting Servlet
##

#
# See Main Servlet on how to protect this servlet.
#
spring.remoting.servlet.hosts.allowed=127.0.0.1,SERVER_IP
spring.remoting.servlet.https.required=false

##
## WebDAV Servlet
##

#
# See Main Servlet on how to protect this servlet.
#
webdav.servlet.hosts.allowed=127.0.0.1,SERVER_IP
webdav.servlet.https.required=false

##
## Admin Portlet
##

admin.default.group.names=
admin.default.role.names=Power User\nUser
admin.default.user.group.names=

admin.mail.host.names=

admin.reserved.email.addresses=
admin.reserved.user.ids=

admin.email.from.name=Joe Bloggs
admin.email.from.address=test@liferay.com

admin.email.user.added.enabled=true
admin.email.user.added.subject=com/liferay/portlet/admin/dependencies/email_user_a
admin.email.user.added.body=com/liferay/portlet/admin/dependencies/email_user_adde

admin.email.password.sent.enabled=true
admin.email.password.sent.subject=com/liferay/portlet/admin/dependencies/email_pas
admin.email.password.sent.body=com/liferay/portlet/admin/dependencies/email_passwo

##
## Alfresco Content Portlet
##

#
# Set this TO true to simulate single sign on for those who do not
have a
# true SSO engine installed. This should never be set in
production because
# of its inherent security hole.
#
alfresco.content.one.step.edit.sso.simulate=false

#
# Set this to the location of the Alfresco server URL.
#

```



```
alfresco.content.server.url=http://localhost:8080

##
## Calendar Portlet
##

#
# Set the list of event types. The display text of each of the
event types
# is set in content/Language.properties.
#
calendar.event.types=anniversary,appointment,bill-payment,birthday,breakfast,call,
club-event,concert,dinner,graduation,happy-hour,holiday,interview,lunch,meeting,mo
other,party,performance,reunion,sports-event,travel,tv-show,vacation,wedding

calendar.email.from.name=Joe Bloggs
calendar.email.from.address=test@liferay.com

calendar.email.event.reminder.enabled=true
calendar.email.event.reminder.subject=com/liferay/portlet/calendar/dependencies/em
calendar.email.event.reminder.body=com/liferay/portlet/calendar/dependencies/email

##
## Document Library Portlet
##

#
# Uncomment the following and set the appropriate values if the
server
# resides on a remote server. Make sure the EJBs are available on
the remote
# server and not on the portal server.
#
#com_liferay_documentlibrary_ejb_DLManager_java.naming.factory.initial=com.evermin
#com_liferay_documentlibrary_ejb_DLManager_java.naming.provider.url=ormi://host/de
#com_liferay_documentlibrary_ejb_DLManager_java.naming.security.principal=admin
#com_liferay_documentlibrary_ejb_DLManager_java.naming.security.credentials=1234

#
# Set the directories where documents are stored. This is now
deprecated
# because documents are stored in Jackrabbit.
#
dl.root.dir=/home/liferay/documentlibrary/root/
dl.version.root.dir=/home/liferay/documentlibrary/vroot/

#
# Set the maximum file size and valid file extensions for
documents. A value
# of 0 for the maximum file size can be used to indicate unlimited
file
# size. A file extension of * will permit all file extensions.
#
# You can map a GIF for the extension by adding the image to the
theme's
# image display and document library folder. The wildcard
extension of *
# will be ignored. For example, the default image for the DOC
extension
# would be found in:
/html/themes/classic/images/document_library/doc.gif.
#
#dl.file.max.size=307200
#dl.file.max.size=1024000
```

```
dl.file.max.size=5120000
dl.file.extensions=.bmp,.css,.doc,.gif,.gz,.htm,.html,.jpg,.js,.pdf,.png,.ppt,.rtf
.tar,.tiff,.tgz,.txt,.xls,.xml,.zip,.odt,.ott,ods

#
# Set the following to true to cache directory views.
#
dl.version.cache.directory.views=true

##
## Image Gallery Portlet
##

#
# Set the maximum file size and valid file extensions for images.
A value
# of 0 for the maximum file size can be used to indicate unlimited
file
# size. A file extension of * will permit all file extensions. Set
the
# maximum thumbnail height or width to 0 to ignore that dimension.
#
ig.image.max.size=307200
ig.image.extensions=.gif,.jpeg,.jpg,.png
ig.image.thumbnail.max.height=50
ig.image.thumbnail.max.width=50

##
## Journal Portlet
##

#
# Set the list of article types. The display text of each of the
article
# types is set in content/Language.properties.
#
journal.article.types=general,news,press-release,test,announcements,blogs

#
# Set this to true so that only the latest version of an article
that is
# also not approved can be saved without incrementing version.
#
journal.article.force.increment.version=false

#
# Set the maximum file size and valid file extensions for images.
A value
# of 0 for the maximum file size can be used to indicate unlimited
file
# size. A file extension of * will permit all file extensions.
#
journal.image.small.max.size=51200
journal.image.extensions=.gif,.jpeg,.jpg,.png

#
# Input a list of comma delimited class names that extend
# com.liferay.portlet.journal.util.TransformerListener. These
classes will
# run in sequence to allow you to modify the XML and XSL before
it's
# transformed and allow you to modify the final output.
#
journal.transformer.listener=com.liferay.portlet.journal.util.TokensTransformerLis
```

```

com.liferay.portlet.journal.util.PropertiesTransformerListener,com.liferay.portlet
#journal.transformer.listener=com.liferay.portlet.journal.util.TokensTransformerLi

#
# Set whether to synchronize content searches when server starts.
#
journal.sync.content.search.on.startup=false

journal.email.from.name=Joe Bloggs
journal.email.from.address=test@liferay.com

journal.email.article.approval.denied.enabled=false
journal.email.article.approval.denied.subject=com/liferay/portlet/journal/dependen
journal.email.article.approval.denied.body=com/liferay/portlet/journal/dependencie

journal.email.article.approval.granted.enabled=false
journal.email.article.approval.granted.subject=com/liferay/portlet/journal/depende
journal.email.article.approval.granted.body=com/liferay/portlet/journal/dependenci

journal.email.article.approval.requested.enabled=false
journal.email.article.approval.requested.subject=com/liferay/portlet/journal/depend
journal.email.article.approval.requested.body=com/liferay/portlet/journal/dependen

journal.email.article.review.enabled=false
journal.email.article.review.subject=com/liferay/portlet/journal/dependencies/email
journal.email.article.review.body=com/liferay/portlet/journal/dependencies/email_a

##
## Journal Articles Portlet
##

#
# Set the available values for the number of articles to display
per page.
#
journal.articles.page.delta.values=5,10,25,50,100

##
## Mail Portlet
##

#
# Uncomment the following and set the appropriate values if the
server
# resides on a remote server. Make sure the EJBs are available on
the remote
# server and not on the portal server.
#
#com_liferay_mail_ejb_MailManager_java.naming.factory.initial=com.evermind.server.
#com_liferay_mail_ejb_MailManager_java.naming.provider.url=ormi://host/default
#com_liferay_mail_ejb_MailManager_java.naming.security.principal=admin
#com_liferay_mail_ejb_MailManager_java.naming.security.credentials=1234

#
# Set the following to false if administrator should not be
allowed to
# change the mail domain via the Admin portlet.
#
mail.mx.update=true

#
# Set the name of the class that implements
com.liferay.mail.util.Hook.
# The mail server will use this class to ensure that the mail and

```

```

portal
# servers are synchronized on user information. The portal will
not know
# how to add, update, or delete users from the mail server except
through
# this hook.
#
# Available hooks are:
#   com.liferay.mail.util.CyrusHook
#   com.liferay.mail.util.DummyHook
#   com.liferay.mail.util.SendmailHook
#   com.liferay.mail.util.ShellHook
#
mail.hook.impl=com.liferay.mail.util.DummyHook

#
# CyrusHook
#
# Set the commands for adding, updating, and deleting a user where
# %1% is the user id. Replace the password with the password for
the cyrus
# user.
#
mail.hook.cyrus.add.user=cyrusadmin password create %1%
#mail.hook.cyrus.add.user=cyrus_adduser password %1%
mail.hook.cyrus.delete.user=cyrusadmin password delete %1%
#mail.hook.cyrus.delete.user=cyrus_userdel password %1%
mail.hook.cyrus.home=/home/cyrus

#
# SendmailHook
#
# Set the commands for adding, updating, and deleting a user where
# %1% is the user id and %2% is the password. Set the home and
# virtual user table information.
#
mail.hook.sendmail.add.user=adduser %1% -s /bin/false
mail.hook.sendmail.change.password=autopasswd %1% %2%
mail.hook.sendmail.delete.user=userdel -r %1%
mail.hook.sendmail.home=/home
mail.hook.sendmail.virtusertable=/etc/mail/virtusertable
mail.hook.sendmail.virtusertable.refresh=bash -c "makemap hash
/etc/mail/virtusertable < /etc/mail/virtusertable"

#
# ShellHook
#
# Set the location of the shell script that will interface with
any mail
# server.
#
mail.hook.shell.script=/usr/sbin/mailadmin.ksh

#
# Set the mail box style that your IMAP server uses. Washington
IMAP uses
# "mail/" whereas Courier IMAP and Cyrus IMAP use "INBOX." as
their mail box
# styles. The mail box style is an IMAP implementation specific
namespace
# that is used in referencing folders.
#
#mail.box.style=mail/
mail.box.style=INBOX.

```

```
#
# Set the name of the Inbox folder. Most IMAP servers use "INBOX"
as the
# folder name. Domino requires "Inbox" as the folder name.
#
mail.inbox.name=INBOX
#mail.inbox.name=Inbox

#
# Set other default folder names.
#
mail.spam.name=Spam
mail.sent.name=Sent
mail.drafts.name=Drafts
mail.trash.name=Trash

#
# The user will be warned once per session to empty their spam if
their spam
# folder exceeds this size. Set the size to 0 to disable any
warnings.
#
mail.spam.warning.size=5120000

#
# The user will be warned once per session to empty their trash if
their
# trash folder exceeds this size. Set the size to 0 to disable any
warnings.
#
mail.trash.warning.size=5120000

#
# Some mail servers do not allow a "." in their user names. Set
the
# following to true to replace all "." in a user name with a "_"
when making
# calls to the mail server.
#
mail.username.replace=true

#
# Set to true to enable SMTP debugging.
#
mail.smtp.debug=false

#
# Input a list of comma delimited email addresses that will
receive a BCC of
# every email sent through the mail server.
#
mail.audit.trail=

#
# Set the maximum file size for attachments.
#
mail.attachments.max.size=3072000

#
# Specify a class name that implements
# com.liferay.portlet.mail.util.multiaccount.AccountFinder.
Another
# implementation could allow the Mail portlet to access multiple
```

```
accounts.  
  #  
mail.account.finder=com.liferay.portlet.mail.util.multiaccount.JferrerAccountFinde  
  
##  
## Message Boards Portlet  
##  
  
    message.boards.email.from.name=Joe Bloggs  
    message.boards.email.from.address=test@liferay.com  
  
    message.boards.email.message.added.enabled=true  
message.boards.email.message.added.subject.prefix=com/liferay/portlet/messageboard  
message.boards.email.message.added.body=com/liferay/portlet/messageboards/depend  
message.boards.email.message.added.signature=com/liferay/portlet/messageboards/dep  
  
    message.boards.email.message.updated.enabled=true  
message.boards.email.message.updated.subject.prefix=com/liferay/portlet/messageboa  
message.boards.email.message.updated.body=com/liferay/portlet/messageboards/depend  
message.boards.email.message.updated.signature=com/liferay/portlet/messageboards/d  
  
##  
## Shopping Portlet  
##  
  
    #  
    # Set the following to true if cart quantities must be a multiple  
of the  
    # item's minimum quantity.  
    #  
    shopping.cart.min.qty.multiple=true  
  
    #  
    # Set the following to true to forward to the cart page when  
adding an item  
    # from the category page. The item must not have dynamic fields.  
All items  
    # with dynamic fields will forward to the item's details page  
regardless of  
    # the following setting.  
    #  
    shopping.category.forward.to.cart=false  
  
    #  
    # Set the following to true to show special items when browsing a  
category.  
    #  
    shopping.category.show.special.items=false  
  
    #  
    # Set the following to true to show availability when viewing an  
item.  
    #  
    shopping.item.show.availability=true  
  
    #  
    # Set the maximum file size and valid file extensions for images.  
A value  
    # of 0 for the maximum file size can be used to indicate unlimited  
file  
    # size. A file extension of * will permit all file extensions.  
    #  
    shopping.image.small.max.size=51200  
    shopping.image.medium.max.size=153600
```

```
shopping.image.large.max.size=307200
shopping.image.extensions=.gif,.jpeg,.jpg,.png

shopping.email.from.name=Joe Bloggs
shopping.email.from.address=test@liferay.com

shopping.email.order.confirmation.enabled=true
shopping.email.order.confirmation.subject=com/liferay/portlet/shopping/dependencies/
shopping.email.order.confirmation.body=com/liferay/portlet/shopping/dependencies/e

shopping.email.order.shipping.enabled=true
shopping.email.order.shipping.subject=com/liferay/portlet/shopping/dependencies/em
shopping.email.order.shipping.body=com/liferay/portlet/shopping/dependencies/email

##
## Translator Portlet
##

#
# Set the default languages to translate a given text.
#
translator.default.languages=en_es

##
## Wiki Portlet
##

#
# Set the name of the default page for a wiki node. The name for
the default
# page must be a valid wiki word. A wiki word follows the format
of having
# an upper case letter followed by a series of lower case letters
followed
# by another upper case letter and another series of lower case
letters. See
# http://www.usemod.com/cgi-bin/wiki.pl?WhatIsaWiki for more
information on
# wiki naming conventions. It is
#
wiki.front.page.name=FrontPage
```

3. system.properties

The `system.properties` file is provided as a convenient way to set all properties for the JVM machine and related system settings. Start your application server with the system property `system.properties.load` set to true to load it. When the server starts, the portal will load `system.properties` and then `system-ext.properties`.

Start your application server with the system property `system.properties.final` set to true if the properties of `system.properties` override all similar command line properties. If set to false, the properties of `system.properties` will be set if and only if those properties are not currently set.

Some application servers require you to set the `file.encoding` and `user.timezone` on startup regardless of `system.properties` because the application server reads these properties before `system.properties` is ever loaded.

Following the default values of the `system.properties` properties are shown:

```
##
## Java
##

#
# The file encoding must be set to UTF-8 in order for the
# internationalization to work correctly.
#
file.encoding=UTF-8

#
# Java uses the underlying operating system to generate images. If
you are
# using Unix and do not start the portal in a X Windows session,
then Java
# will not know how to generate images and you'll get lots of
nasty
# exceptions. Setting this property to true will fix that.
Sometimes this
# property cannot be set dynamically when the server starts and
you'll need
# to edit your start script to include this as a system property.
#
java.awt.headless=true

#
# Set the default language.
#
user.country=US
user.language=en

#
# The time zone must be set GMT so that the portal knows how to
properly
# translate time across time zones.
#
user.timezone=GMT

##
## Servlet Filters
##

#
# If the user can unzip compressed HTTP content, the compression
filter will
# zip up the HTTP content before sending it to the user. This will
speed up
# page rendering for users that are on dial up.
#
com.liferay.filters.compression.CompressionFilter=true
com.liferay.filters.compression.CompressionFilter.encoding=UTF-8

#
# The strip filter will remove blank lines from the outputted
content. This
# will speed up page rendering for users that are on dial up.
#
com.liferay.filters.strip.StripFilter=true
com.liferay.filters.strip.StripFilter.encoding=UTF-8

#
# The layout cache filter will cache pages to speed up page
rendering for
# guest users. Set the refresh time in milliseconds on how often
```



```
the cache
# should refresh. Set it to 0 if the cache never times out.
Default time out
# is set to one hour.
#
com.liferay.portal.servlet.filters.layoutcache.LayoutCacheFilter=true
com.liferay.portal.servlet.filters.layoutcache.LayoutCacheFilter.encoding=UTF-8
com.liferay.portal.servlet.filters.layoutcache.LayoutCacheFilter.refresh.time=3600

#
# The virtual host filter maps hosts to public and private pages.
For
# example, if the public virtual host is www.helloworld.com and
the friendly
# URL is /helloworld, then http://www.helloworld.com is mapped to
# http://localhost:8080/web/helloworld.
#
com.liferay.portal.servlet.filters.virtualhost.VirtualHostFilter=true

#
# The CAS filter will redirect the user to the CAS login page for
SSO. See
# http://www.ja-sig.org/products/cas for more information.
#
com.liferay.filters.sso.cas.CASFilter=false

##
## Logging
##

#
# Set any logger that implements org.apache.commons.logging.Log.
#
#org.apache.commons.logging.Log=org.apache.commons.logging.impl.Log4JLogger

#
# Set the following to true if Log4j complains that it was not
properly
# configured.
#
log4j.configure.on.startup=true

##
## Encryptor
##

#
# Set the security provider class.
#
com.liferay.util.Encryptor.provider.class=com.sun.crypto.provider.SunJCE

##
## HTTP
##

#
# Set the location of the HTTP proxy that the portal will use to
fetch
# external content.
#
#http.proxyHost=192.168.0.200
#http.proxyPort=4480

#
```

```
# These are read for backwards compatibility and only used if the
# "http.proxyHost" and "http.proxyPort" settings are not
specified.
#
#com.liferay.util.Http.proxy.host=192.168.0.200
#com.liferay.util.Http.proxy.port=4480

#
# Set the proxy authentication type.
#
#com.liferay.util.Http.proxy.auth.type=username-password
#com.liferay.util.Http.proxy.auth.type=ntlm

#
# Set user name and password used for HTTP proxy authentication.
#
#com.liferay.util.Http.proxy.username=
#com.liferay.util.Http.proxy.password=

#
# Set additional properties for NTLM authentication.
#
#com.liferay.util.Http.proxy.ntlm.domain=
#com.liferay.util.Http.proxy.ntlm.host=

#
# Set the connection timeout when fetching HTTP content.
#
com.liferay.util.Http.timeout=5000

##
## Format
##

#
# Set the location of the class that implements
# com.liferay.format.PhoneNumberFormat. This class will format
phone
# numbers.
#
com.liferay.util.format.PhoneNumberFormat=com.liferay.util.format.USAPhoneNumberFo
#com.liferay.util.format.PhoneNumberFormat=com.liferay.util.format.IdenticalPhoneN

##
## Axis
##

#
# Urls that match the specified pattern will use the
SimpleHTTPSender. Urls
# that do not match the specified pattern will use Axis' default
HTTPSender.
#
com.liferay.util.axis.SimpleHTTPSender.regexp.pattern=.*mappoint\.net.*

##
## XSS (Cross Site Scripting)
##

#
# Set the pattern to strip text of XSS hacks.
#
com.liferay.util.XSSUtil.regexp.pattern=(?i)<[\\s]*/?script.*?>|<[\\s]*/?embed.*?>
```

```
##
## Upload Servlet Request
##

#
# Set the maximum file size. Default is 1024 * 1024 * 1000.
#
com.liferay.util.servlet.UploadServletRequest.max.size=104857600

#
# Set the temp directory for uploaded files.
#
#com.liferay.util.servlet.UploadServletRequest.temp.dir=C:/Temp

##
## Session Parameters
##

#
# Encrypt session parameters so that browsers cannot remember
them.
#
com.liferay.util.servlet.SessionParameters=true

##
## Lucene
##

#
# Set a timeout to avoid lock errors.
#
#org.apache.lucene.writeLockTimeout=30000

##
## Quartz
##

#
# Make threads daemon so the JVM exits cleanly.
#
org.quartz.threadPool.makeThreadsDaemons=true

##
## Velocity Engine
##

#
# Input a list of comma delimited class names that extend
# com.liferay.util.velocity.VelocityResourceListener. These
classes will
# run in sequence to allow you to find the applicable
ResourceLoader
# to load a Velocity template.
#
com.liferay.util.velocity.VelocityResourceListener=com.liferay.portal.velocity.Ser
com.liferay.portal.velocity.JournalTemplateVelocityResourceListener,\
com.liferay.portal.velocity.DefaultVelocityResourceListener
```

Chapter 2. Customizing Liferay with plugins

Liferay support three types of deployable plugins to customize a Portal installation:

- Themes
- Layout templates
- Portlets

They are all packaged as WAR packages with specific configuration files. This section explains how to deploy one of these plugins.

1. Introduction to Portlets

Portlets are web applications developed for integration within a portal. A portlet can provide any desired functionality: showing the local weather, offering full message boards functionality, providing an interface for back end services, or orchestrating different web services in a SOA architecture.

Liferay comes with several useful bundled portlets and also supports 100% the JSR-168 standard, which allows the portal administrators to deploy any third party portlet developed according to this standard.

After deploying a portlet to Liferay any user that has the roles required by the portlet itself will be able to use it in their private or public home pages, in the private or public website of a community, or in the public website.

2. Introduction to Themes

Themes make it possible to easily switch to different presentations or "look and feel" layers. Within a single .war file, a designer/developer can deliver an integrated package of Velocity or JSPs, Javascript, image, and configuration files that will control all presentation logic and design attributes for a portal community. Liferay Portal comes with a handful of pre-made themes that showcase its versatility:





- Different themes can be assigned to specific user community websites or even to a specific page within the website
- Users can choose a unique theme for their own personal portal page or for each of the pages
- Both Java Server Page (JSP) and Velocity (VM) languages are supported
- Velocity based Themes are hot-deployable as .wars (when supported by the application server)

To select a different theme, go to the **Look and Feel** section in the header bar and choose one of the available themes and a corresponding color scheme. The process is as straightforward as setting a new desktop background in Windows.

3. Introduction to Layout Templates

Layout templates define the areas where a user can place the portlets in a page. By default, Liferay comes with several different templates that are shown in the following picture:



Users of private pages or administrators of communities or public websites can choose the layout to use for each portlet page.

While the default layout options are suitable for most situations there can be situations when you need a very specific portlet window organization, such as website frontpages. To achieve maximum flexibility Liferay allows portal administrators to deploy extra layouts developed either by themselves or by third parties. Developing a layout is easy and can be done in a short amount of time.

Chapter 3. Deployment of Plugins

1. Deployment Options

1.1. Standard Deployment

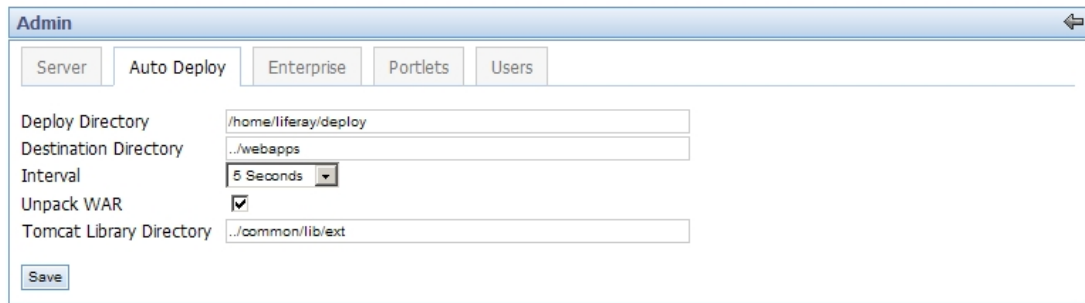
Portlet, themes, and layout templates are all packages as standard Java EE WAR files. For this reason they can be deployed using the standard method for the application server being used.

Note that you will probably have to restart the application server so that the portal registers the new portlet, theme, or layout. To avoid doing a restart follow the instructions in the next section.

1.2. Hot Deployment

Liferay allows you to easily hot deploy layout templates, JSR 168 portlets, and themes. Layout templates allow portlets to be arranged inside the constraints of custom layouts. JSR 168 portlets add functional abilities to the portal. Themes modify the look and feel of the portal. Layout templates, portlets, and themes can be deployed at runtime by utilizing the hot deploy features of Liferay.

1. Go to the drive where you installed your server. Check to see that `/home/liferay/deploy` exists. If it does not exist, create it. You can also customize the location of this directory from the Admin portlet:



The screenshot shows the 'Admin' portlet configuration window with the 'Auto Deploy' tab selected. The configuration includes the following fields:

Deploy Directory	<input type="text" value="/home/liferay/deploy"/>
Destination Directory	<input type="text" value="..webapps"/>
Interval	<input type="text" value="5 Seconds"/>
Unpack WAR	<input checked="" type="checkbox"/>
Tomcat Library Directory	<input type="text" value="..common/lib/ext"/>

A 'Save' button is located at the bottom left of the configuration area.

2. Start the server. The server will now automatically scan for `*.war` files.
3. Download one of the layout templates, sample portlets, or themes to `/home/liferay/deploy`. Click here [<http://www.liferay.com/web/guest/downloads/samples>] for a list of sample portlets, themes and layouts. (Any JSR 168 compliant portlet WAR will work as well.)
4. Install either JBoss+Jetty, JBoss+Tomcat, Jetty, Resin, Tomcat, or WebSphere.
5. If you have already set up the extension environment, you can hot deploy portlets, layouts, or themes by dropping them into `ext/portlets`, `ext/layouttpl`, or `ext/themes` respectively. You can then run **ant deploy** from that directory.

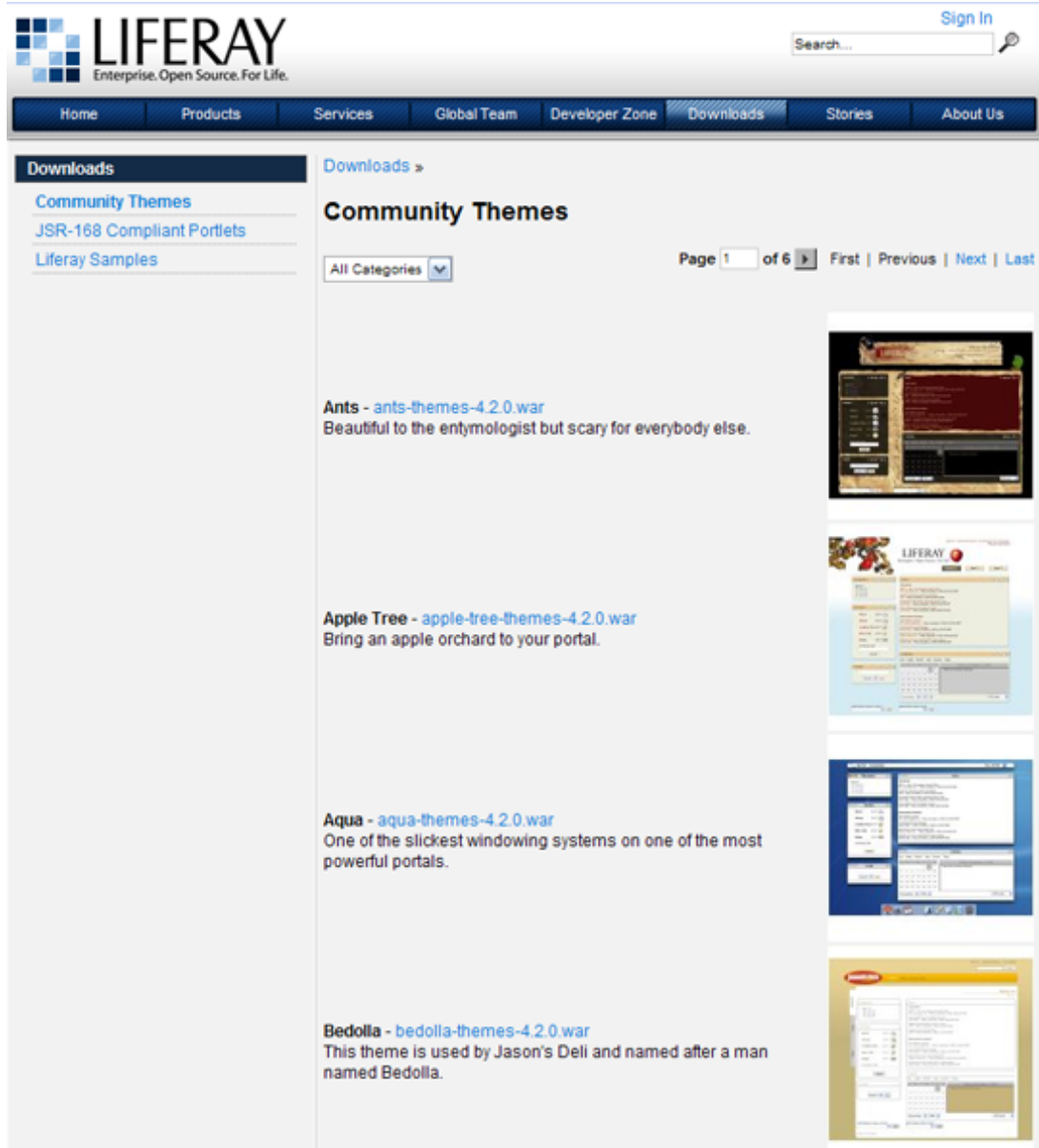
Notes: If the hot deploy feature does not work, make sure that your `*.war` file is a JSR 168 compliant portlet WAR, theme, or layout. (You cannot use themes and layouts designed for Liferay Portal v3.6.1 or layouts and themes that were designed for other portals).

2. Deploying Third Party plugins

2.1. Deploying Community Themes

Along with the prepackaged themes in Liferay Portal, there are many additional themes that are available on Liferay's download page. In this section you will learn how to deploy a community theme onto Liferay Portal.

1. Go to www.liferay.com.
2. Click **Downloads**.
3. Click **Community Themes**.



The screenshot shows the Liferay website's Downloads section. The header includes the Liferay logo, a search bar, and a navigation menu with links for Home, Products, Services, Global Team, Developer Zone, Downloads, Stories, and About Us. The main content area is titled "Community Themes" and features a list of themes with their names, file names, and descriptions. Each theme entry is accompanied by a small thumbnail image of the theme's interface. The themes listed are:

- Ants** - [ants-themes-4.2.0.war](#)
Beautiful to the entymologist but scary for everybody else.
- Apple Tree** - [apple-tree-themes-4.2.0.war](#)
Bring an apple orchard to your portal.
- Aqua** - [aqua-themes-4.2.0.war](#)
One of the slickest windowing systems on one of the most powerful portals.
- Bedolla** - [bedolla-themes-4.2.0.war](#)
This theme is used by Jason's Deli and named after a man named Bedolla.

4. Select a theme. For this demonstration, the Aqua theme will be added. Click on the Breeze theme link.
5. Choose a location to download from and click **Download**.

6. **Save** file.
7. Log into Liferay Portal and sign in as the Administrator.
8. In the Admin Portlet, click on **Auto Deploy**.

The screenshot shows the 'Admin' portlet interface with the 'Auto Deploy' tab selected. The configuration fields are as follows:

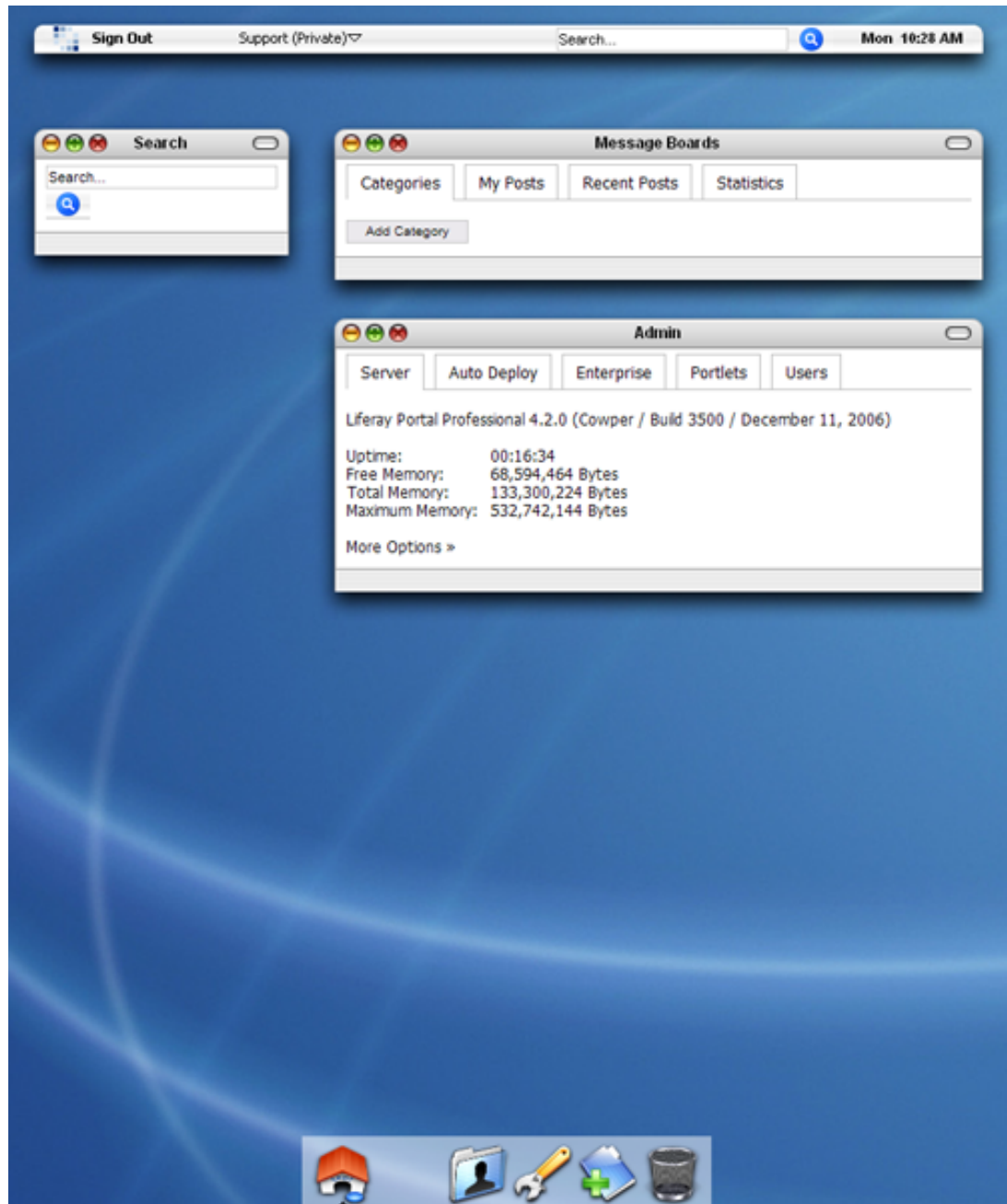
Deploy Directory	<input type="text" value="/home/liferay/deploy"/>
Destination Directory	<input type="text" value="..webapps"/>
Interval	<input type="text" value="5 Seconds"/> <input type="button" value="v"/>
Unpack WAR	<input checked="" type="checkbox"/>
Tomcat Library Directory	<input type="text" value="..common/lib/ext"/>

Below the configuration fields is a **Save** button. Underneath is a section titled **Hot Deploy** with the instruction: "Upload a WAR file to hot deploy a layout template, portlet, or theme." This section contains an empty text input field, a **Browse...** button, and a **Deploy** button at the bottom.

9. Click **Browse** and locate the Aqua theme file.
10. Click **Deploy**.
11. To deploy the Aqua theme on the page, click **Page Settings**.
12. Click **Look and Feel**.



13. Select the Aqua theme to deploy. The figure below shows the Aqua theme deployed.

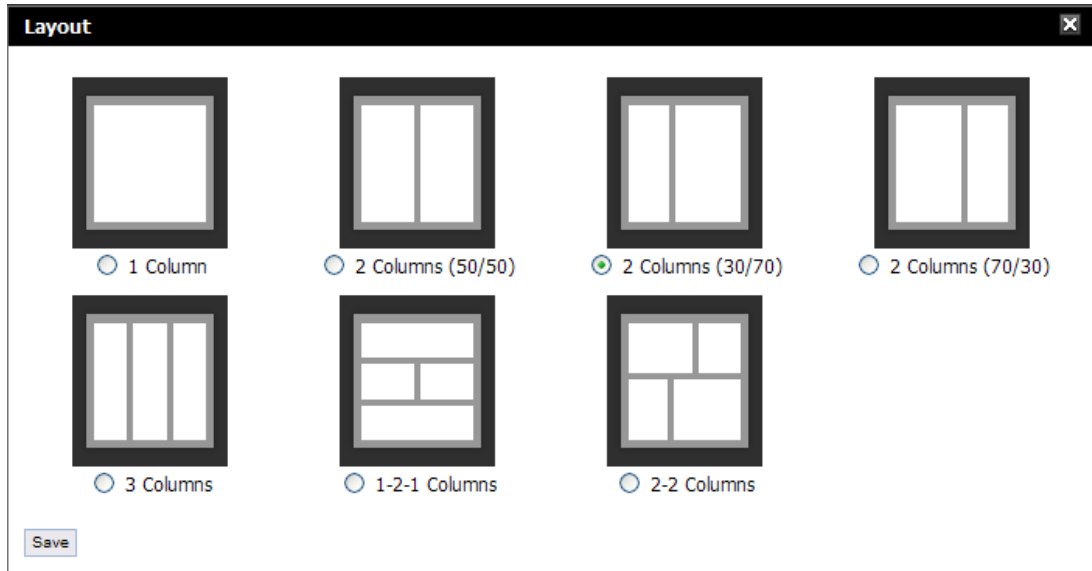


2.2. Adding Layout Template

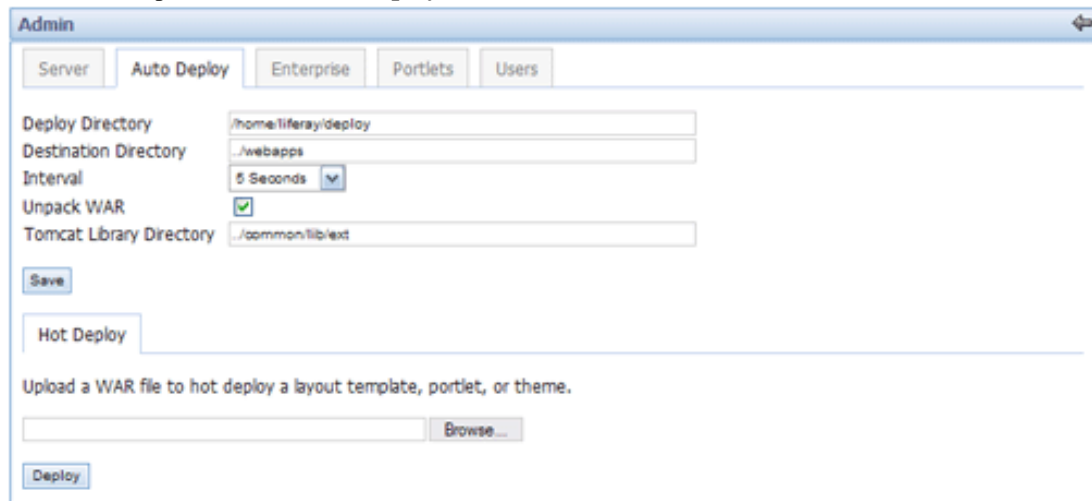
The pages of Liferay Portal are organized according to the layout templates used. Seven different layout templates are provided with the option for users to write and add new ones. This section will provide instructions for adding a sample layout template.

1. Begin by logging in as the Administrator.
2. To see a list of layout templates that comes prepackaged with Liferay Portal, click **Layout**.

- Seven templates are available.



- To add an additional template, go to Liferay’s download page located at <http://www.liferay.com/web/guest/downloads>.
- Click on **Liferay Samples**.
- The sample layout template that will be added contains a 2-1-2 layout. The first row contains two columns of portlets, the second row contains one column of portlets, and the third row contains two columns of portlets. Click on the **Sample Layout Template**.
- Select a location to download from and **Save** the file.
- After the file has been saved, log back in as the Administrator.
- In the Admin portlet, click **Auto Deploy**.



- Click **Browse** to locate the file, and click **Deploy**.
- To verify that the new template has been added, click on **Layout**.

12.The 2-1-2 layout template will appear in the list.