Intralibrary Installation Guide: Version 2.7

Author: Intrallect Ltd
Company: Intrallect Ltd

Table Of Contents

1. Prerequisites

1.1. Java Environment

This version of intraLibrary takes advantage of the many improvements to the Java Runtime Environment offered by Java 1.4 over Java 1.3. For this version of intraLibrary to function, it must be running under Java 1.4.

The full *sdk* must be installed and available on your server. The *jre* alone is not enough. The Java 1.4 sdk is available for many platforms here:http://java.sun.com/j2se/

IntraLibrary has been tested[1] on the Windows, Linux and Solaris environments available from Sun and also on Apple's Java 1.4 implentation for Apple OS X.

1.2. Database

Although intraLibrary is designed to run using almost any *ANSI-SQL* compliant database, we **strongly** recommend using MySQL version 4.0 for this version of intraLibrary. It is anticipated that we will be in a better position to support other database vendors with subsequent revisions to the intraLibrary application.

1.3. Application Server

If you are using Tomcat as your application server then we **strongly** recommend that you use version 4.1.x. intraLibrary will run on any J2EE complaint Java Webserver adhering to the Servlet 2.3 specification and the JSP 1.2 specification, but deployment of the intraLibrary web application will require a significant amount of configuration not covered in this document.

The Tomcat 4.1.x server is available in two versions. The full version and an *LE* version. We recommend that you use the *LE* version which has been designed to use the XML parser bundled with Java 1.4.

If you intend deploying intraLibrary on anything other than Tomcat, then follow the instructions from your application server vendor on how to deploy a web application. Pay particular attention to the details of setting up a datasource described later in this document. For help in deploying to a Servlet Container other than Tomcat 4.1.x contact a member of our support team.

NOTE: We (and others!) have been experiencing strange behaviour when running Tomcat as a service on the Windows platform. We cannot currently recommend running Tomcat as service using the Tomcat service installation tool. However some of our customers have had more success by creating a service manually. If you'd like help with this, contact a member of our support team.

1.4. Installation Package

If you're reading this, then you probably already have the intraLibrary 2.7 installation package.

This package contains:

- IntraLibrary web application archive - *intralibrary.war*
- config directory containing:
  - Global intraLibrary settings - *(intralibrary.properties)*
  - authentication2 directory containing:
    - The default authentication process - *(authentication_chain_descriptor)*
    - A sample set of LDAP authentication properties - *(ldap_authenticator.properties)*
    - A sample set of SQL authentication properties - *(sql_authenticator.properties)*
  - emailTemplates directory containing default language directory ("en") which contains the following:
    - The text of the publish notification email sent to administrators - *(adminPublishNotificationEmail.txt)*
    - The text of the notification email sent to administrators or librarians when annotations are added to an object - *(annotationNotificationEmail.txt)*
    - The text of the publish notification email sent to users - *(userPublishNotificationEmail.txt)*
    - The text of the notification email sent to users when an object is moved - *(stageChangeEmail.txt)*
    - The text of the notification email sent to named external users when an object is moved into a certain stage - *(stageChangeExternalUserEmail.txt)*
    - The text of the notification email sent to the object owner when an object is moved into a certain stage - *(stageChangeObjectOwnerEmail.txt)*
- dataSourceLib directory containing:

  - MySQL database driver3 - *(mysql-connector-java-3.0.11-stable-bin.jar)*
  - Apache Commons DBCP library - *(commons-dbcp.jar)*
  - Apache Commons Pool library - *(commons-pool.jar)*
  - Apache Commons Collections library - *(commons-collections.jar)*
- endorsed directory containing:
  - Xalan library - *(xalan-2.5.2.jar)*
  - Xerces library - *(xerces-2.6.0.jar)*
  - XML API's for Xerces library - *(xml-apis.jar)*
- dbscripts directory containing:
  - Main database setup script - *(intralibrary.sql)*
  - Add initial user setup script - *(createusers.sql)*
  - UK LOM Core Application Profile script- *(uklomcore0p2.sql)*
  - CANCore Application Profile script - *(cancore.sql)*
  - Sample MySQL conf file - *(my.ini)*
- docs directory
- licences directory for third party components
- sample webapp context file - *(sample_context.xml)*

1.5. Internet Connection

The intraLibrary web application needs access to the internet to perform some XML validation that occurs when using certain areas of functionality within intraLibrary.

The intraLibrary web application also needs to be able to send email from the application server. A valid, working and reachable smtp server must be defined. We will define this later.

1.6. Pre-Installation

This installation guide assumes that both an application server and a database server have been correctly installed. We would recommend that you check both are working correctly before moving onto the installation of intraLibrary.

2. Installation

2.1. IntraLibrary Web Application

The basis of the intraLibrary web application is the *intralibrary.war file*. You can put this archive anywhere on the server, but we'd recommended it to be on the same physical volume as the application server (Tomcat).

Once the web application is in place we can start to configure the properties to match your system

2.2. Configuring Properties

In this version of intraLibrary all configurable aspects (global properties, authentication, email content) are stored within a single *config* directory. This directory can be found at the root level of the installation package and should be copied to the filesystem somewhere (this can be anywhere on the file system). We **strongly** recommend that the directory is created outside of the webapp directory to ease the upgrade process in the future. Once this directory has been copied we can go on to edit the global properties.

There are a number of properties that need to be defined in the **config/intralibrary.properties** file. The file supplied with the installation package contains some sample values.

The mandatory entries that need to be edited are:

- **returnAddress**

  Return address for any emails that are sent out. This should be set so the recipient can identify which instance of the application the mail refers to. For example *intralibrary1@yourcompany.com*

- **repositoryName**

  This is the name used to identify this instance of intraLibrary

- **adminEmail**

  Address for system admin (report bugs, broken links etc).

- **mailHost**

  This needs to be a valid smtp server that can process outgoing mail from the application.

- **idPrefix**

  This needs to be a globally unique id used to identify this instance of intraLibrary. Top avoid potential conflicts we suggest using an reverse internet naming scheme. For example *com.your-company.intralibrary1*

The optional entries that need to be edited are:

- **proxyHost**

  This is the host name or ip address of the proxy that the application needs to be routed through.

- **proxyPort**

  This is the port that the proxy is listening on.

- **errorEmailAddresses**

  Comma seperated list of addresses to be notified when an the application logs a user out due to an internal error. We recommend that this be a member of our support team. However, you may like to keep track of these errors yourself and perhaps forward them to our support team if you feel it necessary.

- **entryPage**

  This property overrides the default entry page for all users. If this is not set, then all users will be taken to the browse library page. If you wish users to be taken to their upload area after login, then set this property to *upload*. This has no effect for users who do not have access to an upload area.

- **serverTimeout**

  This value (in seconds) sets how long before a user session is timed-out. By default this is set to 1800(30 mins).

- **objectCacheSize**

  This maintains the upper limit for objects stored in the application cache. The default is 250. Change this to higher values for improved performance on machines with a lot of physical memory or reduce it for machines with hardly any free memory.

- **populateCacheOnStartup**

  The default is true. If you have a large cache size, then startup time for the application will be slowed down while the cache is populated. During debugging or testing, it may be useful to set this property to false. Note if the cache isn't populated at start up then the initial browsing and searching of object will run slowly.

- **reindexOnStartup**

  The default is true. If you have a lot of objects in the repository, then startup time for the application will be slowed down while all the metadata for each object is indexed. During debugging or testing, it may be useful to set this property to false.

- **logLevel**

  Determines the level, and quantity, of logging messages that will be output. Valid values are DEBUG, INFO, WARN, ERROR, FATAL

- **browseIndexPage**

  Allows an external page to be specified to replace the page shown when the user selects 'Browse Library'

- **supportURL**

  Specifies a url to form a link on the help section index page. This can be a web based url or a mailto link.

2.3. Deploying the Licence

The licence file will be provided seperately to the intraLibrary application, if you don't yet have a licence please contact a member of our support team.

To deploy the licence and ensure that it will be picked up by your intraLibrary installation the file *intralibrary-licence.txt* should be placed at the root level of the *config* directory which you created in the previous section.

2.4. Configuring the intraLibrary context

For the rest of this section we will refer to the root directory of your Tomcat installation as ***TOMCAT_HOME***.

The intraLibrary web application needs to be *registered* to allow the application server to run it. We do this by creating a *context* in the server.xml file. This file can be found at ***TOMCAT_HOME*/conf/server.xml**.

Included in this install package is a sample *context.xml*. You can make the changes for your specific installation and then copy this *context* element into your server.xml file. The *context* element needs to go at the same level as any other *contexts*, inside the *host* element and before any *logger* elements.

There are only a couple of attributes of the *context* element that need to be changed. For all other parts, the defaults should be fine. More information about configuring server.xml and and contexts can be found here:http://jakarta.apache.org/tomcat/tomcat-4.1-doc/config/context.html

- **docBase**

  This attribute indicates where on your server the *intralibrary* web archive is located. Remember, we recommend that this should be on the same physical volume as the application server itself.

  example: docBase="/disk1/tools/repositories/intralibrary.war"

- **path**

  This attribute maps the docBase to the part of the url the user will use to locate intraLibrary.

  example: if the path is *'/intralibrary'* and the base url of your application server is *'http://www.company.com'*, the user would access intraLibrary by accessing the url *'http://www.company.com/intralibrary'*

- **Environment name="configDir"**

  This attribute indicates where on your server the *config* folder is located, you should only change the *value* part of this entry as any other changes will prevent the application from running successfully. Remember, we recommend that this folder lives outwith the web application.

  example: <Environment name="configDir" type="java.lang.String" **value="/Volumes/projects/intralibrary/config"** override="true" />

2.5. Configuring a Datasource

NOTE: The following instructions are written specifically for creating a *datasource* using Tomcat 4.1.x, the MySQL 4.0 database engine and the database driver we supply. If you are not using this configuration, you will need to follow the guidelines set out by your application server vendor for your specific database. If you need more help with this then please feel free to contact a member of our support team.

To enable the communication between the application and the database, we need to create a *datasource*. We do this by making changes to the *context* we created in the section above.

Inside the *context* element discussed above, you'll find a *Resource* element and a *ResourceParams* element. You can leave the *Resource* element as it is, but you will need to make changes to the *ResourceParams* element. Listed below are the elements and attributes that you will need to edit to setup the *datasource*. For all other parts, the defaults should be fine. More information about configuring a datasource using Tomcat can be found here:http://jakarta.apache.org/tomcat/tomcat-4.1-doc/jndi-datasource-examples-howto.html

ResourceParams

- **url**

  This is the connection string that the application would use to connect to the database. This will vary depending on the database engine. Your database driver documentation will provide details of the connection string needed. If you are using MySQL and the database driver we supply, the example below will work fine as long as the database server is on the same machine as the application server and can by connected to as *localhost*. If the database server is not on the same machine as the application server, you will have to change the *localhost* part of the url, to the name, or IP address of the server the database resides on.

  MySQL, by default, runs on port 3306. If your database engine runs on a different port then the port part of the url will obviously have to be changed.

  The last part of the url before the '?' is the database name.

example: *jdbc:mysql://localhost:3306/intralibrary?autoReconnect=true*

- **maxIdle**

  This is the maximum number of idle database connections that you wish to have in the database connection pool. For unlimited connections, set to 0.

  example: *100*

- **maxActive**

  This is the maximum number of database connections in the pool4. Set to 0 for no limit.

  example: *500*

- **driverClassName**

  This is the fully qualified name of the JDBC Driver class. This will be supplied by the vendor of the database driver you are using. If youa re using the database driver we have supplied then the example value below will be fine.

  example: *com.mysql.jdbc.Driver*

- **maxWait**

  This is the maximum time to wait for a database connection to become available in ms, in this example 10 seconds. To wait indefinitely set this value to -1. An error will occur if the application has to wait for a time exceeding this limit.

  example: *10000*

- **username**

  This is the username that the application will use to connect to the database. This user must have read and write access to the database whose name is defined in the database connection url defined above.

  example: *intrallect*

- **password**

This is the password for the user defined above.

example: *intrallect*

- **factory**

  Unless you are implementing a different database pool to the one supplied, leave this entry as is.

The final part of configuring the *datasource* is to copy the following files from the install package *dataSourceLib* directory to *TOMCAT_HOME/common/lib*:

If you are propmted to overwrite any existing files, go ahead. You only live once!:

- commons-collections.jar
- commons-dbcp.jar
- commons-pool.jar
- mysql-connector-java-3.0.6-stable-bin.jar

2.6. Configuring Tomcat endorsed Directory

NOTE: The following instructions are written specifically for using Tomcat 4.1.x.

Although Tomcat already contains XML libraries, we need to make sure that the ones that are loaded are the ones that we need. To enable us to do this we make use of the ability to add jar files to an *endorsed* directory.

To achieve this you should copy the following files from the install package *endorsed* directory to *TOMCAT_HOME/common/endorsed*:

- xerces-2.6.0.jar
- xalan-2.5.2.jar
- xml-apis.jar

3. Database Setup

3.1. Creating the Database

We need to create a database for the application to use. We do this by executing the following sql statement:

create database intralibrary;

you do not need to use the name *intralibrary* but the following instructions will assume you have.

Next we need to create a user for the application to connect to the database as and give that user the correct permissions. Here we are using the username/password of intrallect/intrallect. This can be set to anything you like but must match the values that were used when setting up the datasource.

grant all on intralibrary.* to intrallect identified by 'intrallect';

grant all on intralibrary.* to intrallect@'%' identified by 'intrallect';

5 ***grant all on intralibrary.* to intrallect@'localhost' identified by 'intrallect';***

3.2. Populating the Database

We now need to create the database tables and populate these tables with the initial values that the application needs.

In the dbscripts folder, there are two sql files which needs to be run to set up the database. It is important that they are run in this order:

1. intralibrary2p1.sql
2. createusers.sql

There are a number of ways that this can be done. The easiest and most reliable is to connect to the database and from the database command line issue the following command:

source *absolute/path/to/file;* e.g. source /home/intralibrary/dbscripts/intralibrary2p1.sql;

(obviously replacing 'absolute/path/to/file' with the actual file path!)

If you wish to add support for UKLomCore and CANCore application profiles, you should also run the *uklomcore.sql* and *cancore.sql* files in this directory.

3.3. Database Configuration

Although it is outside the scope of this documentation to go into a detailed discussion of database configuration, when using MySQL, the following advice may be useful.

MySQL configuration lives in a file named either *my.cnf* or *my.ini*. Its name and location depends on the installed environment. The following parameters (and numerous others) can be set in this file and effect performance of intraLibrary and should be investigated. We've included some sample values which we've found to work well but your environment may differ. For more information see here:http://www.mysql.com/doc/en/MySQL_Database_Administration.html

- set-variable = max_connections=100000
- set-variable = max_allowed_packet=20M
- set-variable = wait_timeout=360

4. Confirming Installation

We should now be ready to restart the application server and confirm that the intraLibrary application is running.

The first thing we need to do is to make sure we can see or read the output from the application server. Depending on how you set up the *context*, this output could go to one of a few places. By default, Tomcat writes all its logging to files inside *TOMCAT_HOME/logs*. If you have used the supplied *context.xml* file then the output file you should be interested in is *catalina.out*.

Before you start the server make sure you have included the licence file in the correct place.

If you start the server now, you should see some output to the log file that resembles the text below.

```
**********************************************

populating learning object cache with 0 objects...
finished populating learning object cache

**********************************************
application version:2.7(72010)
database server version:4.0.18-standard
database url:jdbc:mysql://localhost:3306/intralibrary?autoReconnect=true
database driver:MySQL-AB JDBC Driver
max database connections(0=unlimited):0
max upload file size:49Mb
servlet container:Apache Tomcat/4.1.18-LE-jdk14
java version:1.4.2_03
----------------------------------------------
configDir location:/Volumes/projects/intralibrary/config
http proxy:no proxy used
mail host:mail.company.com
admin email:admin@company.com
error email:errors@company.com
return email:intralibrary1@company.com
id prefix:com.comapany.intralibrary1
repository name:intralibrary1
object cache size:250
entry page:browse
debug:false
reindexOnStartup:true
----------------------------------------------
total memory assigned:26Mb
maximum memory assigned:112Mb
----------------------------------------------
licence assigned to:company
max contributors:10
current contributors:1
expires:
----------------------------------------------
IntraLibrary started May 01, 2006 10:23:12 AM
**********************************************
```

Assuming that you have no errors in this output then congratulations, you have successfully installed intraLibrary 2.7.