explicanto Distribution Platform Server Documentation

explicanto Developer Documentation for the Server of the Distribution Platform

Table of contents

1. How to build the explicanto Distribution Server

Prerequisites – Before the BEAS-explicanto software can be installed, you must install the prerequisites software described below.

New Installation – Once the additional products are installed, you can install and configure the BEAS-explicanto software.

2. Prerequisites

BEAS-explicanto requires the following products to work properly:

1. 1.Java2sdk 1.4.2_07 – you can download a version of this package from http://java.sun.com
2. 2.Jakarta ant-1.6.2 – you can download a version of this package from http://jakarta.apache.org
3. 3.Liferay 3.2.0+tomcat edition – you can download a version of this package from http://www.liferay.com/cms/servlet/HOME-INDEX
4. 4.PostgreSQL 8.0.1 – you can download a version of this package from http://www.postgresql.org
5. 5.The PostgreSQL JDBC driver pg74.215.jdbc3.jar – you can download a version of the package from http://jdbc.postgresql.org/download.html#jars

You must follow the steps:

1. Install java2sdk (i.e. c:/j2sdk-1.4.2_07). Set JAVA_HOME variable to root installation folder.
2. Install Jakarta ant-1.6.2 (i.e. c:/ant-1.6.2). Set ANT_HOME variable to root installation folder.
3. Set up your path to include %ANT_HOME%\bin and %JAVA_HOME%\bin.
4. Install PostgreSQL with the default options.
5. Install Liferay-3.2.0+tomcat (i.e. c:/liferay-3.2.0). Set LIFERAY_HOME variable to root installation folder.

3. New Installation

Download the explicanto-pack project from the SVN repository. This will be the EXPLICANTO_HOME from now on.

This archive contains.

- EDS, this will be the EDS_HOME
- PORTLET, this will be PORTLET_HOME
- YAWIKI, this will be YAWIKI_HOME

- The etc folder from EDS, PORTLET and YAWIKI contains the configuration files for deploying the applications
- The build.properties file
- The build.xml file

Create two databases in PostgreSQL, 'lportal' and 'explicanto', both should be created on the same DB Server.

Create an user in PostgreSQL to use to give liferay so it can access the databases.

You must set your JAVA_HOME in the LIFERAY_HOME/bin/Catalina.bat file (i.e. SET JAVA_HOME=c:/j2sdk-1.4.2_07).

Go into the LIFERAY_HOME/conf/Catalina/localhost (change the localhost with whatever host you set up for Tomcat, if you do) and make the necessary changes in the liferay.xml file (url, driver class, username, password).

4. Library Dependencies

There are several jar that have to be copied into different folders:

Into the EDS_HOME/Web-Inf/lib folder (create it if it doesn't exist):

- ADLParser.jar
- antlr.jar
- aopalliance.jar
- cglib-full-2.0.2.jar
- cmidatamodel.jar
- commons-beanutils.jar
- commons-collections-2.1.1.jar
- commons-dbcp-1.2.1.jar
- commons-digester.jar
- commons-fileupload.jar
- commons-lang-1.0.1.jar
- commons-lang-2.1.jar
- commons-logging.jar
- commons-logging-api.jar
- commons-pool-1.2.jar
- commons-validator.jar
- concurrent-1.3.3.jar
- connector.jar
- debug.jar
- dom4j-1.4.jar

- ehcache-0.9.jar
- ejb.jar
- hibernate2.jar
- informa.jar
- itext-1.02b.jar
- jaas.jar
- jakarta-oro.jar
- jasperreports-0.6.7.jar
- jcommon-1.0.0-pre2.jar
- jcs-1.0-dev.jar
- jdbc2_0-stdext.jar
- jdom.jar
- jfreechart-1.0.0-pre2.jar
- jgroups-2.2.7.jar
- jlo.jar
- jsonrpc.jar
- jta.jar
- junit-3.8.1.jar
- jZonic.jar
- jZonic-addon.jar
- jZonic-cache.jar
- log4j-1.2.8.jar
- odmg-3.0.jar
- optional.jar
- oscache-2.1.1.jar
- pg74.215.jdbc3.jar
- quartz.jar
- reload-diva.jar
- reload-dweezil.jar
- reload-jdom.jar
- reload-moonunit.jar
- reload-support.jar
- servlet-api.jar
- sitemesh-2.1.jar
- spring.jar
- struts.jar
- velocity-1.4.jar
- velocity-dep-1.4.jar
- xalan-2.4.0.jar
- xerces-2.4.0.jar

- xjavadoc-1.0-SNAPSHOT.jar
- xml-apis.jar

Into the PORTLET_HOME/Web-Inf/lib folder (create it if it doesn't exist):

- jstl.jar
- log4j-1.2.8.jar
- util-java.jar
- util-taglib.jar

Into the YAWIKI_HOME/Web-Inf/lib folder (create it if it doesn't exist):

- activation.jar
- commons-logging.jar
- cos.jar
- crimson.jar
- EXML.jar
- gnu-regexp-1.1.4.jar
- hsqldb.jar
- javax.servlet.jar
- jaxp.jar
- jconfig.jar
- jlo.jar
- jmxri.jar
- jmxtools.jar
- jZonic.jar
- jZonic-addon.jar
- jZonic-cache.jar
- lucene-1.2.jar
- mail.jar
- mailapi.jar
- perpojo.jar
- pop3.jar
- rsslibj-1_0RC2.jar
- smtp.jar
- soap.jar
- velocity-dep-1.3.1-rc2.jar
- xalan.jar

5. Configuration

There are different properties files that have to be modified:

In the EXPLICANTO_HOME folder there is the build.properties config file which contains different properties that have to be set to reflect you're local file system. Here you must set the folowing properties

- deploy.path
- eds.tomcat.contexts.path
- wiki.tomcat.contexts.path
- eds_portlet.tomcat.contexts.path
- tomcat.conf.context
- liferay.lib.ext
- psql
- psql.username
- psql.password
- psql.database

In the EDS_HOME folder there is a **explicanto-system.properties** file which has to be modified to reflect your database setings

Also in EDS_HOME/etc and PORTLET_HOME/etc folder there are log configuration files **log4j.properties** which can be modified to change the log level (INFO, DEBUG, etc).

Run ant full-deploy from the EXPLICANTO_HOME folder - this will create the explicanto database tables (explicanto database must be created and empty before running this task), it will restore the lportal database (lportal database must be created and empty before running this task) and then it will deploy the apps, and copy the need it jars in the liferay lib directory.

Attention, the files from the repository are from a Linux deployment, you should change them for a Windows one.

6. Starting

Start LIFERAY – from the LIFERAY_HOME/bin/startup.bat.

Now the installation is finished and you can start the application. Open Internet Explorer browser and open the following URL: http://youraddress:yourport.

The application will start and the main page should be displayed. Login in the explicanto Administration portlet with automatically created test account: username=a and password=a.