explicanto

Table of contents

1. About

1.1. explicanto Open Source

1.1.1. Overview

explicanto is a integrated E-Learning, Knowledge Management and Knowledge Distribution Platform.

explicanto helps to capture, transform and distribute knowledge in a quick, efficient and organized way. This is performed independently of time and place.

The Closed-Loop feedback process allows for monitoring of the critical success factors in knowledge delivery and acceptance.

It consists of two Java-based product components:

- Client-Server based **Authoring Platform** - Java Rich Client (based on eclipse RCP) and a Java Server (Java EE, Web Services, JDBC/ RDBMS)
- Web based **Distribution Platform** - Java Server (Java EE, Portlets, JDBC/ RDBMS)

explicanto has been developed by Beck et al. Services GmbH.

Note:

explicanto and the explicanto logo are registered trademarks of Beck et al. Services GmbH.

1.1.2. Characteristics

- A simple product with versatile means for implementation
- High acceptance because of its user-friendliness
- Knowledge: collected only once, unlimited utilization
- Optimum effect through its performance monitoring system

1.2. explicanto's Mission Statement

A company’s success greatly depends on one factor: Knowledge.

It is vital that employees properly understand internal processes and applications to work efficiently. They need accurate and in-depth information about the company’s products so that the best possible business success can be achieved. Only with sufficient know-how will they be able to provide high-quality service. By following this principle, any company with an efficient knowledge exchange and training will always be one step ahead of the competition.

However, this challenge is enormous: New competitors are constantly entering global markets. Product life cycles are becoming shorter. Laws are continuously changing. Thus, if a business wants to overcome today's challenges, it will need to implement an efficient knowledge transfer process.

Know-how transfer should not become a one-way process. It is important to ensure that information is understood correctly. The costs and benefits of investing in a sufficient knowledge transfer process must be measured continuously.

What is a Best-Practice for turning the success factor knowledge into performance?

explicanto will provide the solution. It is designed to elevate knowledge to the optimum, and is customized to meet your exact individual business needs.

explicanto helps to *capture, transform* and *distribute* knowledge in a *quick, efficient* and *organized* way. Moreover, this is performed independently of time and place.

The Closed-Loop feedback process allows for monitoring of the critical success factors in knowledge delivery and acceptance. Therefore, explicanto will ensure that your business remains competitive.

1.3. Benefits

1.3.1. Fast collection

The experts in each department (editors) can easily and quickly collect, transform and update their knowledge. To achieve this, they neither require programming knowledge nor do they need to staff external specialists. Content that has already been collected (e.g. office documents, videos and illustrations) is easily integrated.

1.3.2. Efficient teamwork

Teams and individuals can prepare information at the same time, unrestricted by time and location. To-Do lists, Status Tracking and Notifications provide transparency. Automated processing steps result in efficient teamwork.

1.3.3. Central availability of knowledge

All information is consolidated and stored in a central database. Didactic structures and templates support this preparation. Changes within the central database can be performed with the push of a button regardless of the delivery medium, and are instantly available through all distribution channels.

1.3.4. Customized delivery medium and information channels

The prepared content can be adapted into a multitude of formats – from PDF, Flash or Java files to HTML and Handhelds. This makes it possible to focus on the most diverse communication channels – whether it be presentations, manuals, classroom- or online-training.

1.3.5. Very user-friendly

Relevant information is available to each target group in a concrete format. It is quick and easy to access and can be used at any time. The content is visually appealing and is didactically prepared. explicanto takes the individual user needs into consideration. This provides for user satisfaction.

1.3.6. Goal-orientated communication

explicanto helps to prevent the accumulation of unnecessary content. Users are able to provide their individual input about which knowledge areas they perceive to be the most relevant. They are informed of new or changed content automatically. This enables users to receive up-to-date information for any topic of their choice.

1.3.7. Optimizing Knowledge Management through feedback

Learning content must not only be distributed, but understood as well. Through reports and other feedback channels, the editor is able to monitor the extent to which the individual user has understood the provided information. Comments and suggestions for further improvement that users contribute continuously optimize this central knowledge database.

1.3.8. Easy to expand

explicanto can initially be used on a small scale, and if desired can be expanded later. Thus, explicanto can be adapted depending on the company’s growth.

1.3.9. High security measures

explicanto has adopted very reliable security measures. Only authorized people are allowed to view and change content. A state-of-the-art encryption technology ensures secured transmission – even beyond the company’s internal network.

1.3.10. Future-proof solutions

explicanto exclusively uses future-proof technology (e.g. Java Enterprise Edition), which is independent of specific platforms. Therfore, the long-term use of investment is ensured.

1.4. Use Cases

1.4.1. Sales

Successful selling requires an excellent information management system. Even before market entry, your explicanto experts prepare their individual knowledge whether it be product details, sales strategies or pricing queries. This knowledge information will then be communicated to all sales personnel and sales partners.

At the same time explicanto provides necessary **Kick-Off material**, which includes **presentations** as well as **online-training** courses, available within a company's internal network or over the internet. If customers require more specific product information, it can be found on your CD. Updating knowledge is easy, since all information is stored in a standardized database. Monitoring performance can be performed while using explicanto. Due to these features, it is guaranteed that all information is up-to-date and properly formatted.

1.4.2. Customer Support and Service

Customer Service is focused on increasing customer satisfaction and at the same time reducing cost per case. explicanto can put both of these goals into practice. explicanto facilitates further training of internal and external employees at no extra cost. New employees can be trained most efficiently. If operations are outsourced, explicanto will ensure that the business partner is sufficiently qualified to achieve the expected quality standards. The **Closed-Loop feedback** system enables monitoring of these processes. A further advantage: The necessary service know-how is not distributed amongst various contacts, but is consolidated and always available.

1.4.3. IT department

In a business environment, changes are constantly taking place within the IT department, from introducing new software to updating existing applications. It is therefore vital that employees continuously expand their knowledge. Even before implementing changes, the required knowledge can be collected using explicanto.

Following this scheme, it is possible to efficiently train users before and during an introduction phase. For every setting the most appropriate method can be used, either through **classroom-** or **online-training**. Updates are collected and communicated efficiently within the system. Since access to this knowledge reservoir is permanent, workers will always receive guaranteed support. Solutions for frequently recurring problems are immediately recorded with the help of explicanto and are available to all users. Therefore, the optimum benefits of new software or processes can be taken advantage of.

1.4.4. Human Resource Management

Communication of internal and external regulations results in a need for training and in particular the training of new employees. A comprehensive information management policy contributes enormously to increasing employee satisfaction and motivation levels. It also contributes to cost reduction. explicanto offers a valuable solution to reduce costs: It provides a knowledge store which can be used for any type of training – whether it be basic information concerning the operational workflow, or regulations for responsible actions. **Updating** can be done in real-time and communicated immediately. The **sustainability** of these measures can be monitored via reports and learning tests.

1.5. Solutions

1.5.1. Industry/ Engineering

Increase in sales efficiency and service quality through multimedia-based product information.

1.5.2. Service Management

Successful implementation of service management processes (ITIL,COBIT) through project-related information, communication and training.

1.5.3. Financial Services

Increase in agent commitment and sales efficiency through prompt and concentrated knowledge transfer to the point-of-sale.

1.5.4. Defense & Security

Meet the challenges of global security policy through permanent exchange of and access to up-to-date knowledge.

2. Developer Documentation

2.1. explicanto Documentation

The explicanto Developer documentation is organized according to the product components:

1. **Authoring Platform**
   - Authoring Server
   - Authoring Client
2. **Distribution Platform**
   - Distribution Server

2.2. Authoring Platform

2.2.1. explicanto Authoring Platform Documentation

2.2.1.1. Overview

The Explicanto Authoring Platform is a client/server system composed from the Explicanto Server and the Explicanto Authoring Client. In order to start the system you will need:

- A JRE 1.5 (or higher)
- A Tomcat 5.0.30 (or higher) distribution for the Explicanto Server. The server is a web application deployed as a WAR file and runs inside the Tomcat container
- A relational database (Postgres, Mysql) for the Explicanto Server.
- Eclipse 3.1 (or higer) IDE. This is required in order to build the client. For the server you will need ant. The client is build around RCP from Eclipse and you will have to use Eclipse to import and build it.

You can fetch and build the server and the client in any order but the recommanded method is to build and run the server first and to build the client last. This is because the client communicates with the server using a web-service and the dependent classes should regenerate after the server is started. The current distribution includes the generated code against the latest version of the server and you should have to problem in using it. If there are changes made on the server then the client code has to be regenerated.

2.2.2. explicanto Authoring Platform Client Documentation

2.2.2.1. How to build the Explicanto Server

The Explicanto Server is generated from several source libraries. Each library is available as an Eclipse project on the svn and should be imported into the Eclipse workspace. After the import is complete and you add the external libraries you can use the Ant script to build the server project.

A **build** of the server will result in a war file that should be uploaded into a Tomcat distribution. You need to have Tomcat 5.0.30 or higher installed. (It is possible the server will run on different application servers but this has not been tested yet - we are happy to put here additional information for various configuration if you succeeded in working with them).

The first step is to fetch all the projects from the trunk/authoring/server folder in svn repository. The projects to be obtained are:

- Libraries
- Deploy
- Bea Environment
- Bea Util
- Vidya Base
- VidyaCommon
- VidyaServer

Even if the order does not matter it is best to start with the Libraries to be able to fill in all the dependencies. The external jars will be distributed in several places:

- /Libraries
- /VidyaServer/CourseTransformator/WEB-INF/lib (create it if it doesn't exist)
- /VidyaServer/ServerLibraries (create it if it doesn't exist)
- /VidyaServer/ServerLibraries/climp (create it if it doesn't exist)
  NOTE: the climp import funcitionality is optional as described below
- /VidyaServer/ServerLibraries/reports (create it if it doesn't exist)

The workspace and projects should look like in the folowing images (libraries from all the folders are automaticaly added to the project's build path):

To build the explicanto server, there is an Ant script in the Deploy project that should be used.
The folowing steps must be covered to perform the build:

1. In the build.xml file from the Deploy project set the value of the **workspacehome** property to the location of youre eclise workspace. Also the JAVA_HOME evironment variabile must be set.
2. Create an Ant Build run configuration from Run->External Tools
3. In the run configuration select the build.xml file from the Deploy projects as the buildfile

4. In the targets tab of the run configuration select the folowind targets in the given order: prepare, compileWebservice, cleanup
5. In the VidyaServer/xpcdb.properties you have to set the database connection settings
6. In the VidyaServer/explicantoServer.properties you can set the course export information. Set the **course_target_location** to the **ROOT/Courses** folder of you're using the Tomcat instalation.
7. Run the ant configuration
8. Deploy the war file into your Tomcat. The war file can be found in the Deploy/build/explicanto folder

Before running the server you will have to create a database. You can select a mysql or postgresql server and use the database creation scripts from the Deploy/tools folder. After you create the database you should initialize the data and update your xpcdb.properties file with the database location and connection credentials.

2.2.2.2. Database setup

For the database you can choose between mysql and postgresql. Both databases are fully supported and have creation scripts. In order to create a new database you have to:

1. Install or start the database engine
2. Create a new user named xpc_user with the password xpc_pwd (of course you can choose any username or password you wish, this is only an example!)
3. Create a new database named xpcdb with the xpc_user having full rights (ownership)
4. Run /Deploy/tools/database-clean-db.sql script (database is mysql or postgresql)
5. Run /Deploy/tools/createInitialContent.sql
6. Run /Deploy/tools/settings-database.sql
7. Run /Deploy/tools/initial.sql

2.2.2.3. External Libraries

Dependencies for the project Libraries:

- activation.jar
- catalina-ant.jar
- cglib2.jar
- commons-collections-3.0.jar
- commons-dbcp.jar
- commons-lang.jar
- commons-logging.jar
- commons-pool.jar
- concurrent.jar
- connector.jar

- dom.jar
- dom4j.jar
- ehcache.jar
- jaas.jar
- jaxb-api.jar
- jaxb-impl.jar
- jaxb-libs.jar
- jax-qname.jar
- jaxrpc-api.jar
- jaxrpc-impl.jar
- jaxrpc-spi.jar
- jboss-cache.jar
- jboss-common.jar
- jboss-jmx.jar
- jboss-system.jar
- jce.jar - it comes with the JRE also!
- jcs.jar
- jgroups.jar
- jta.jar
- junit.jar
- log4j.jar
- looks-1.2.1.jar
- mail.jar
- namespace.jar
- odmg.jar
- optional.jar - in the recent versions of ant this has been splited in several jars.
- oscache.jar
- proxool.jar
- relaxngDatatype.jar
- saaj-api.jar
- saaj-impl.jar
- sax.jar
- swarmcache.jar
- xalan.jar
- xerces.jar
- xercesImpl.jar
- xml-apis.jar
- xsdlib.jar
- xsltc.jar

Dependencies for the VidyaServer/CourseTransformator/WEB-INF/lib folder:

- altrmi-common-0.9.2.jar
- altrmi-registry-0.9.2.jar
- altrmi-server-impl-0.9.2.jar
- altrmi-server-interfaces-0.9.2.jar
- asm-1.4.2.jar
- avalon-framework-api-4.1.5.jar
- avalon-framework-impl-4.1.5.jar
- batik-all-1.5.1.jar
- bsf.jar
- bsf-2.3.0.jar
- castor-0.9.5.3-xml.jar
- cocoon-2.1.5.1.jar
- cocoon-batik-block.jar
- cocoon-bsf-block.jar
- cocoon-fop-block.jar
- cocoon-linkrewriter-block.jar
- cocoon-xsp-block.jar
- commons-cli-1.0.jar
- commons-collections-3.0.jar
- commons-httpclient-2.0-final.jar
- commons-io-1.0.jar
- commons-jexl-1.0-beta-1-20040113.jar
- commons-jxpath-20030909.jar
- commons-lang-2.0.jar
- commons-logging-1.0.3.jar
- excalibur-component-1.2.jar
- excalibur-event-api-1.1.jar
- excalibur-event-impl-1.1.jar
- excalibur-i18n-1.1.jar
- excalibur-instrument-1.0.jar
- excalibur-instrument-manager-1.0.jar
- excalibur-instrument-manager-interfaces-1.0.jar
- excalibur-io-1.1.jar
- excalibur-logger-1.1.jar
- excalibur-monitor-1.0.2.jar
- excalibur-naming-1.0.jar
- excalibur-pool-1.2.jar
- excalibur-sourceresolve-1.1.jar
- excalibur-store-1.0.jar
- excalibur-testcase-1.2.jar

- excalibur-xmlutil-1.0.jar
- fop-0.20.5.jar
- groovy-20040415.114632.jar
- jakarta-bcel-20040329.jar
- jakarta-oro-2.0.8.jar
- jakarta-regexp-1.3.jar
- javacApi-0.9.jar
- javacImpl-0.9.jar
- jcs-1.0-dev-20040516.jar
- jdtcore-2.1.3.jar
- jisp-2.5.1.jar
- log4j-1.2.8.jar
- logkit-1.2.2.jar
- pizza-1.1.jar
- rhino1.5r4-continuations-20040228.jar
- util.concurrent-1.3.3.jar
- xalan-2.6.0.jar
- xercesImpl-2.6.2.jar
- xml-apis.jar
- xml-commons-resolver-1.1.jar
- xreporter-expression-20030725.jar
- xsaxon8.jar

Dependencies for VidyaServer/ServerLibraries:

- bsf.jar
- c3p0-0.8.5.2.jar
- commons-io-1.0.jar
- hibernate2.jar
- jdbc2_0-stdext.jar
- mysql-connector-java-3.1.6-bin.jar
- postgresql-8.0-310.jdbc3.jar
- xalan-2.6.0.jar

Depedencies for the VidyaServer/ServerLibraries/climp:

- axis.jar
- climp.jar - This jar can be found in this folder. It comes with the distribution an it is optional. If it is not added to the server than an error will be shown when an course export is started with climp enabled.
- FTPProtocol.jar
- ibmjsse.jar

- jdom.jar
- roster.jar

Dependencies for the VidyaServer/ServerLibraries/reports folder:

- commons-beanutils-1.5.jar
- commons-collections-2.1.jar
- commons-digester-1.7.jar
- itext-1.3.1.jar
- jasperreports-1.2.0.jar
- jdt-compiler-3.1.1.jar
- tools.jar - This jar is obtained from the installed JDK.

Please pay attention to the libraries names - usually they are without numbers and are referred like that from the projects. If you are using a library with version numbers then you can either remove it or update the project (locally).

2.2.2.4. Common Build Problems

If the workspace shows errors in all the projects except Libraries and Deploy then you should do the folowing:

1. Remove Bea Environmet, Bea Util, VidyaBase, VidyaCommon, VidyaServer projects from the workspace WITHOUT removing them from the filesystem. Select DO NOT DELETE CONTENT in the message box that appears when you try to delete.
2. Import them back again from the current workspace using Import->Existing Projects Into Workspace
3. Perform a full workspace rebuild

Make sure that the Java compiler lever is set to 1.4 or higher to avoid errors coming from the **assert** keyword. (there are present in the Bea Util project)

2.2.2.5. TODO

- Specify for each library a minimal version and how to get it
- Specify a build from the sources, with regeneration of the model objects
- Write a hacking howto
- Eliminate dependencies (integrate into code)
- Howto sign with your own keystore/certificates

2.2.3. explicanto Authoring Platform Server Documentation

2.2.3.1. How to build the Explicanto Client

To build the Explicanto Authoring Client you have to cover the folowing steps.

Fetch the project from explicanto svn (trunk/authoring/client/ExplicantoClient)

The explicanto Authoring client is a RCP (Rich Client Plaform) application developed using Eclise IDE and you need to have Eclipse 3.1 (or higher) installed in order to build it. You can download it from External ref

ATTENTION! The ExplicantoClient it is NOT compatible with any Eclipse version above 3.1.1. If you wish to run it on newer versions you'll have to do some changes in the code (no support offered yet).

Import the ExplicantoClient project into the eclipse workspace.

All the necessary external libraries must be added. If you already have the server dependencies then most of them are already present. These have to be copied into the lib folder of the project and are the folowing:

- activation.jar
- commons-logging.jar
- jax-qname.jar
- jaxrpc-api.jar
- jaxrpc-impl.jar
- jaxrpc-spi.jar
- log4j.jar
- mail.jar
- plastic-1.1.2.jar
- saaj-api.jar
- saaj-impl.jar
- xerces.jar
- xml-apis.jar
- itext.jar
- namespace.jar

Additionaly you will need the xored library used to edit html content. It can be requested from xored.com. Currently the website is under construction, the link will be updated asap. You will have to copy the files into your Eclipse's features and plugins folders and restart Eclipse.

If all these libraries don't appear in the project class path then they must be manually added using Java Build Path of the projects properties. The folowing screenshot shows how the libraries should look like:

In the root of the project there is the explicanto.properties file witch contains the address of the Explicanto Server. Modify this for the server's location

You can run the Explicanto Client with Run->Eclipse Applicantion feature or you can export the project with Export->Deployable plugins and features and run it externaly

2.2.3.2. TODO

- Specify for each library a minimal version and how to get it
- Explain ExplicantoClientFeature usage (optional, for webstart)
- Write a hacking howto

2.3. Distribution Platform

2.3.1. explicanto Distribution Platform Documentation

2.3.1.1. Overview

The explicanto Distribution Server is the component used to expose the courses to the learners (end users) and to receive feedback from them. The course distribution and management is done using portlets () which can be embedded in any portlet container.

The basic explicanto implementation is using Liferay portal for running the portlets.

2.3.2. explicanto Distribution Platform Server Documentation

2.3.2.1. 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.

2.3.2.2. 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.

2.3.2.3. 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).

2.3.2.4. 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

2.3.2.5. 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.

2.3.2.6. 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.

3. Site Features

3.1. Site Linkmap Table of Contents

This is a map of the complete site and its structure.

- explicanto Open Source ____________________ *site*
    - About ____________________ *about*
        - Overview ____________________ *overview* : Welcome to explicanto Open Source
        - Mission Statement ____________________ *mission_statement* : Knowledge – the true success factor
        - Benefits ____________________ *benefits* : 10 Reasons for using explicanto
        - Use Cases ____________________ *use-cases* : 10 Reasons for using explicanto
        - Solutions ____________________ *solutions* : explicanto Solutions
    - Developer Documentation ____________________ *documentation* : explicanto