LINKED ENVIRONMENTS FOR ATMOSPHERIC DISCOVERY

MyLead Release V1.2 Installation Guide

Project Title: MyLead
Document Title: myLEAD Release V1.2 Installation Guide
Organization: Indiana University, Distributed Data Everywhere (DDE) Lab
Date: March 24, 2006
Contact: Yiming Sun (yimsun@cs.indiana.edu)
Authorship: Sangmi Lee Pallickara, Scott Jensen, and Yiming Sun

1 Introduction 2
1.1 myLEAD License 2
2 System Specification 3
2.1 Prerequisite Products 3
3 Installation Instructions 4
3.1 Installing Prerequisite Products 4
3.1.1 Installing MySQL v5.0.18 4
3.1.2 Installing Globus Toolkit WS-Core v3.2.1 4
3.1.3 Installing Jakarta Tomcat Server v4.1 5
3.1.4 Installing OGSA-DAI v6.0 OGSI 5
3.2 myLEAD Server 7
3.2.1 Setting Environment Variables 7
3.2.2 Setting the PATH environment variable to include the Java and ANT executables 7
3.2.3 Installing MyLEAD Server 7
3.2.3.1 Creating the Database 8
3.2.3.2 Deploying the myLEAD Factory in OGSA-DAI 9
3.2.3.3 Installing The Stored Procedures 9
3.2.3.4 Installing the LEAD Schema and myLEAD Activity Schemas 10
3.2.3.5 Installing The Jar File 12
3.2.4 Launching myLEAD Server 12
3.3 Installation of the myLEAD Client Service 14
3.3.1 Setting Environment Variables 14
3.3.2 Copying Required Files 14
3.4 Installation of the myLEAD Agent Service 15
3.4.1 Configuration of the MyLead Agent Service 15
3.4.2 Run the MyLead Client Service 16
3.4.3 Compile MyLead Agent Service 16
4 Testing the Installation 17

1 Introduction

This document provides an overview of myLEAD system requirements and how to install the myLEAD system with its sub-components. The document covers:

- System Requirements:
  - System specifications
  - Prerequisite products
- Installation Instructions:
  - Prerequisite products
  - MyLead Server
  - MyLead Client Toolkit
  - MyLead Agent Service
- Test Instructions

1.1 myLEAD License

The file doc/myLEAD-Licence.txt within the source distribution directory contains the product license. Make sure that you read this license and accept its conditions before continuing.

2 System Specification

myLEAD is designed to work on – and has been tested under following platforms:

- Redhat Linux 3.2.3

2.1 Prerequisite Products

myLEAD is built on top of few existing products. The myLEAD system performs only under proper installation and configuration of these products.

- MySQL v5.0.18
- MySQL Connector/J Version 3.1.11 (or higher)

Please see http://www.mysql.com to download software and information.

- OGSA-DAI v6 OGSI

Please see http://www.ogsadai.org.uk where either source or binary distributions are available.

- Globus Toolkit WS-Core v3.2.1

Please visit http://www.globus.org/toolkit to download software and detail information.

- Java

JDK 1.4.2 or later. Please see http://java.sun.com.

- ANT Build tool

To compile myLEAD source distribution with included ant build file, you need:
Jakarta Ant V.1.6.5 or later. See http://jakarta.apache.org/ant

- Jakarta Tomcat Server v4.1.31

Please see http://jakarta.apache.org/tomcat/index.html to download software.

3 Installation Instructions

In this section the process of installation and building myLEAD v1.2 is described. myLEAD release contains three components:

- Prerequisite products
- MyLead server
- MyLead client toolkit
- MyLead agent service

3.1 Installing Prerequisite Products

3.1.1 Installing MySQL v5.0.18

Please refer to the installation instruction included with the MySQL package or the MySQL manual which is available at http://www.mysql.com.

Note: on certain linux platform, the client hostnames are not resolved correctly which may prevent a client from connecting to the database using JDBC. The typical error message is:

```
mysql error: [host] is not allowed to connect to this MySQL server.
```

If this is the case, please connect to the mysql as root and do the following:

```
GRANT ALL PRIVILEGES ON *.* TO ‘[user]’@’[host]’ identified by
‘[password]’
```

where [user] is the username used to access the database, [host] is the same host shown in the error message above, and [password] is the password used by the [user] to connect to the database.

In addition, it is recommended to increase the limits on the maximum number of file descriptors allowed for the user running MySQL. Edit the following file:

```
/etc/security/limit.conf
```

Add the following two lines to increase the number to 65535, assuming MySQL database is run under the username mysql:

```
mysql    soft   nofile    65535
mysql    hard   nofile    65535
```

3.1.2 Installing Globus Toolkit WS-Core v3.2.1

Please refer to the Globus Toolkit installation guide which is available from http://www.globus.org

3.1.3 Installing Jakarta Tomcat Server v4.1

Please refer to http://jakarta.apache.org/tomcat/tomcat-4.1-doc/RUNNING.txt for the installation of Tomcat Server.

3.1.4 Installing OGSA-DAI v6.0 OGSI

Please download the package from http://ogsadai.org.uk/downloads/archive and refer to the included installation guide. It is assumed that while the OGSA-DAI package is being deployed under Tomcat, the Service Factory is named

```
ogsadai/MyLeadGDSF
```

And this name is referenced later in this document for the Service Factory URL. If you have chosen a different name, please change it accordingly.

Note: the website for downloading xmldb.jar mentioned in the OGSA-DAI installation guide no longer exists; however, this jar file is included in the xindice distribution.

Note: if the version of java you are using is v1.4.2_05 or newer, you need to obtain the latest Xalan jar files from http://xml.apache.org/xalan-j/downloads.html and place them in the endorsed directory. If not, you may get an illegal access exception that may look like the following:

```
java.lang.IllegalAccessError: tried to access field
org.apache.xpath.compiler.FunctionTable.m_functions from class
org.apache.xml.security.Init
```

The Xalan jar files include the following three:

- `xalan.jar`
- `xercesImpl.jar`
- `xml-apis.jar`

for myLEAD server, you should place the files in the following directory

```
<TOMCAT_HOME>/common/endorsed
```

where <TOMCAT_HOME> is the installation directory of the tomcat server. For running the client, you should place those files in the following directory

```
<JAVA_HOME>/jre/lib/endorsed
```

where <JAVA_HOME> is the installation directory of Java. However, on certain platforms the Java Run Time (JRE) may be installed in a separate directory, and if that is the case, the Xalan jar files should ALSO be placed in the following directory

```
<JRE_DIRECTORY>/lib/endorsed
```

for example, on a machine running Microsoft Windows XP, the <JRE_DIRECTORY> may look like the following:

```
C:\Program Files\Java\jre1.4.2_08
```

Note: if you are using the latest XML security jar file instead of the xmlsec.jar provided with the package, you may encounter an exception that looks like the following

```
java.lang.VerifyError: Cannot inherit from final class
```

if that's the case, please download the source code of the latest XML Security library from http://xml.apache.org/security/download.html, unzip the packages, and remove the keyword "final" from the line "public final class" in the following java file:

```
src/org/apache/xml/security/signature/XMLSignature.java
```

After the modification, please recompile the package and place the modified xml security jar to the proper location as indicated in the OGSA-DAI installation guide.

3.2 myLEAD Server

The following abbreviations are used,

- <JAVA_HOME> - the path on which the JDK is installed
- <ANT_HOME> - the path on which ANT is installed
- <TOMCAT_HOME> - the path on which the Tomcat Server will be installed

3.2.1 Setting Environment Variables

To set environment variables, please edit your shell file.
For example, to set an environment variable JAVA_HOME:
For csh users, type:

```
setenv JAVA_HOME “/l/jdk1.4.2_04”
```

For other users, type:

```
export JAVA_HOME “/l/jdk1.4.2_04”
```

3.2.2 Setting the PATH environment variable to include the Java and ANT executables

```
export PATH=$PATH:$ANT_HOME/bin:$JAVA_HOME/bin
```

3.2.3 Installing MyLEAD Server

The MyLEAD server installation consists of the following files:

- leaddai.jar, which is the myLEAD library.
- mylead1.2.txt, which is the database schema definition file.
- myLEADSP1.2.tar.gz, which contains the stored procedures used by myLEAD.
- LeadActivities.tar.gz, which contains the XML schema files for the myLEAD activities added to OGSA-DAI.
- MyLead-ActivityMap.txt, which contains changes to the OGSA-DAI activity map.
- LeadTypes.tar.gz, which contains the LEAD schema files and additional type files used by the myLEAD activity schemas.
- log4j.properties, which is used by log4j.
- xpp3-1.1.3.4.B.jar, which is the library for the XPP Pull Parser.
- ogsadai-activities-Indiana2.jar, which is a synchronous version of the OGSA-DAI deliver-to-stream activity.

3.2.3.1 Creating the Database

The first step to installing the myLEAD server is to create the database. While the mylead1.2.txt script will define the tables needed for myLEAD, the database must first be created using the MySQL client. Assuming the MySQL database is running, the MySQL client can be accessed by typing the following command at the system prompt:

```
mysql –u [user] –p
```

For a new installation of myLEAD, create the database by typing the following at the `mysql>` prompt in the MySQL client:

```
CREATE DATABASE mcs_lead;
```

At this point the newly created database does not contain any tables or data. The mylead1.2.txt script can be used to create the necessary tables by typing the following at the system prompt:

```
SOURCE /full/path/to/mylead1.2.txt
```

Deleting The Database

If an earlier version of myLEAD is installed, the existing database should be backed up and then deleted (dropped) before using the above command to create the new database.

The database can be backed up to a text file using the following MySQL command:

```
mysqldump –-add-drop-table –add-locks –c –K –u <user> -p mcs_lead > <file>
```

This will create <file> with all of the table definitions and data. The existing database can then be dropped using the following command:

```
DROP DATABASE IF EXISTS mcs_lead;
```

Now that the database has been created, a user must be defined that has the rights needed to access the database from OGSA-DAI. In the MySQL client, type the following commands to create a new account named `myleaduser` for the GDSF (the GDSF is created in the next step):

```
GRANT ALL ON mcs lead.* TO 'myleaduser'@'%' IDENTIFIED BY 'myleadpw';
GRANT ALL ON mcs_lead.* TO 'myleaduser'@'localhost' IDENTIFIED BY
'myleadpw';
GRANT SELECT ON mysql.proc TO 'myleaduser'@'localhost' IDENTIFIED BY
'myleadpw';
```

This creates a new user named `myleaduser` that has all rights on the new `mcs_lead` database from any host. However, it does not have the `GRANT` option, so the new user cannot create additional accounts. The `SELECT` permissions on `mysql.proc` are needed for the stored procedures that are used by the myLEAD server code. You may want to use a password that is harder to guess – just be sure to enter that password when prompted for the database password during the factory deployment process.

3.2.3.2 Deploying the myLEAD Factory in OGSA-DAI

The next step is to deploy the OGSA-DAI factory that will create Grid Data Service (GDS) instances for myLEAD. Documentation on deploying a factory can be found at:

```
http://www.ogsadai.org.uk/docs/R6.0/doc/ogsi/deploy/DeployGDSF.html
```

In the directory where you unpacked the binary OGSA-DAI version 6.0 (OGSI) installation, run the following command to start the deploy process:

```
ant cliDeployFactory
```

This process will prompt you with the following series of questions:
For the data resource type, enter 2 for MySQL.
For the resource vendor, press the `Enter` key to use the 'MySQL' default.
For the product version, enter 5.0.18 (or later if on a different version).
For the driver class, press the `Enter` key to use the 'org.mysql.jdbc.Driver' default.
For the resource driver specific URL, enter 'jdbc:mysql://localhost:3306/mcs_lead.
For the credential prompt, press Enter for no credentials.
For the database user ID enter 'myleaduser'.
For the database password enter 'myleadpw'.
For the relative path enter 'ogsadai/MyLeadGDSF'
The system will prompt for the driver jar files to be downloaded. This was already done in an earlier step in the installation process, so press the `Enter` key to continue.

When the factory deployment process is done, the configuration files for the service will be located in the following directory:

```
<CATALINA_HOME>/webapps/ogsa/WEB-INF/etc/_ogsadai_MyLeadGDSF
```

3.2.3.3 Installing The Stored Procedures

The stored procedures used by myLEAD are contained in the `myLEADSP1.2.tar.gz` tarball. To install the stored procedures, first unzip the tarball, which creates a directory named `SP` containing all of the stored procedures plus a script named `LoadStoredProc.sh` that can be used to load the procedures. Each stored procedure can be loaded individually from the system prompt in the SP directory using the following command:

```
mysql –u [user] –D mcs_lead --password=[password] < spXXXXX.txt
```

The `LoadStoredProc.sh` script contains the above line for each stored procedure, but you may need to change the username and password for your environment. The script assumes that the user is “root” and has “mylead" as the default for the password. After running the script at the system prompt, be sure to remove your password to avoid compromising your database security.

3.2.3.4 Installing the LEAD Schema and myLEAD Activity Schemas

In OGSA-DAI, there is a separate activity schema for each activity that a user can perform against a database using OGSA-DAI. In myLEAD we have extended this set of activities with activities specific to the functionalities of the myLEAD catalog. When a perform document is sent to the server, OGSA-DAI validates the perform document against these activity schemas; the myLEAD activity schemas are contained in the `LeadActivities.tar.gz` tarball. Unzipping this file will create a directory named `activities` that contains 17 XML schema files needed for the myLEAD activities. These files should be copied to the following directory on the server:

```
<TOMCAT_HOME>/webapps/ogsa/schema/ogsadai/xsd/activities
```

OGSA-DAI uses an activity map file to map both standard and custom activities to both their schema and their implementation in the server code. This mapping is maintained in the `activityConfigRelational.xml` file in the following directory (assuming you entered "`ogsadai/MyLeadGDSF`" as the relative path when deploying the factory):

```
<TOMCAT_HOME>/webapps/ogsa/WEB-INF/etc/_ogsadai_MyLeadGDSF
```

This file consists mainly of an "`activityMap`" element that groups activities by their behavior (delivery activities, transform activities, etc.). Just above the closing tag for the `activityMap` element, insert the definitions for the myLEAD activities. The XML for these changes can be copy/pasted into the activity map from the file named `MyLead-ActivityMap.txt`. In addition to the myLEAD activity schemas in the tarball, there is an additional schema that OGSA-DAI has provided for a synchronous version of the deliver-to-stream activity. In the activity map, the entry for the `deliverToStream` activity must be updated. This activity is in the "`Delivery activities`" section of the map and should be changed to the following:

```
<activity name="indianaDeliverToStream"
      implementation="uk.org.ogsadai.activity.delivery.IndianaDeliverToStreamActivity"
      schema="indiana_deliver_to_stream.xsd" />
```

Since LEAD communicates metadata via the LEAD Metadata Schema (LMS), the LMS plus some supporting myLEAD types must be made available in OGSA-DAI for the myLEAD activities loaded above. These schema files are contained in the `LeadTypes.tar.gz` tarball that creates a directory named `LeadTypes` when unzipped. If myLEAD is being added to an OGSA-DAI installation where the `general_types.xsd` schema file has already been modified for other purposes, then please read the sidebar below regarding `general_types` before installing this file - otherwise copy it and all of the other files in the `LeadTypes` directory to the following directory on the server:

```
<TOMCAT_HOME>/webapps/ogsa/schema/ogsadai/types
```

These schema files assume that myLEAD is running on the standard myLEAD port, 10081. If you are running myLEAD on the standard Tomcat port (8080), or any port than 10081, then both the `general_types.xsd` and `LeadFGDC.xsd` schema files will need to be updated to reflect the port you are using.

In the `general_types.xsd` schema, there are the following schema imports near the top of the schema. Modify the highlighted port addresses to reflect the port where myLEAD is running on your server:

```
<!--   ********************** LEAD Import **********************   -->
<!-- The FGDC import must be included before the LEAD and LEADElement -->
<!-- imports to allow those schemas to set schema location based only -->
<!-- on the schema name and not a full path since it will already     -->
<!-- have been loaded in the FGDC import.                             -->
<!-- If myLEAD is running on a port other than 10081, then the port   -->
<!-- for the schema location must be updated to that port number.     -->
<xsd:import namespace="FGDC" schemaLocation="http://localhost:10081/ogsa/schema/ogsadai/types/LeadFGDC.xsd"/>
<xsd:import namespace="LEADElements" schemaLocation="LEADElements.xsd"/>
<xsd:import namespace="LEAD" schemaLocation="http://localhost:10081/ogsa/schema/ogsadai/types/LEAD.xsd"/>

<!--    ******************** myLEAD Import ********************     -->
<!-- The following import requires that the mylead_types.xsd file be  -->
<!-- in the specified directory and that the ml namespace declaration -->
<!-- be included above: ml=myLEADTypes                                -->
<!-- If myLEAD is running on a port other than 10081, then the port   -->
<!-- for the schema location must be updated to that port number.     -->
<xsd:import namespace="myLEADTypes" schemaLocation="http://localhost:10081/ogsa/schema/ogsadai/types/
mylead_types.xsd"/>
```

The `LeadFGDC.xsd` schema contains only imports and includes for components of the LMS in order to make them available to the myLEAD activities. Each line of this file is a schema import or include similar to that shown above for the `general_types.xsd` schema - the port should be updated on each line.

Modifying the OGSA-DAI `general_types.xsd` Schema

The `general_types.xsd` file in OGSA-DAI contains the main OGSA-DAI schema. When validating perform documents, OGSA-DAI combines all of the standard activities and custom activities (such as those added for myLEAD) with this schema to determine if the perform document contains a valid activity. When a custom activity requires elements from an external schema such as the LMS, the `general_types.xsd` file must be modified to import those schemas and declare their namespaces. The version of this file contained in the `LeadTypes.tar.gz` tarball contains all of the original schema plus namespace declarations and schema imports needed for myLEAD. If you have NOT already modified this schema for other custom OGSA-DAI activities, then you can copy in the version modified for myLEAD.

If you have already customized the `general_types.xsd` schema, then you will need to add the imports shown above and also add the following highlighted namespace declarations to the schema element at the beginning of the file:

```
<xsd:schema targetNamespace="http://ogsadai.org.uk/namespaces/2005/03/types"
  xmlns:tns="http://ogsadai.org.uk/namespaces/2005/03/types"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ml="myLEADTypes"
  xmlns:LEAD="LEAD"
  xmlns:FGDC="FGDC"
  xmlns:LE="LEADElements"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
```

3.2.3.5 Installing The Jar File

All of the server code is contained in the myLEAD `leaddai.jar` file. In addition to this jar file there are three other jar files that must also be installed. The first is a jar provided by the OGSA-DAI team that contains a synchronous version of the OGSA-DAI deliver-to-stream activity. This activity is contained in the `ogsadai-activities-Indiana2.jar` file. The second jar file that must be installed is the XPP3 Pull Parser, which is contained in the `xpp3-1.1.3.4.B.jar` file. The last of the three additional jar files that must be installed is the jar file for MySQL Connector/J that you installed when installing MySQL. All four of these files should be copied to the following directory:

```
<TOMCAT_HOME>/webapps/ogsa/WEB-INF/lib
```

The OGSA-DAI deliver-to-stream activity also requires minor changes to the `web.xml` file that defines the OGSA-DAI servlets and their mappings to OGSA-DAI activities. The `web.xml` file is located in the following directory:

```
<TOMCAT_HOME>/webapps/ogsa/WEB-INF
```

This file consists of a set of `<servlet>` and `<servlet-mapping>` elements. The following elements need to be added to this file following the last servlet and servlet-mapping elements respectively:

```
<servlet>
    <servlet-name>DeliverToStreamServlet</servlet-name>
    <display-name>OGSA-DAI Deliver To Stream Servlet</display-name>
    <servlet-class>
        uk.org.ogsadai.activity.delivery.IndianaDeliverToStreamServlet
    </servlet-class>
</servlet>
```

```
<servlet-mapping>
    <servlet-name>DeliverToStreamServlet</servlet-name>
    <url-pattern>/servlet/DeliverToStreamServlet</url-pattern>
</servlet-mapping>
```

`The myLEAD Server Installation is Now Completed!`

Note: we have also provided a file log4j.properties which can be used by the Tomcat server for debugging. Place this file somewhere and set the following environment variable:

```
setenv CATALINA_OPTS –Dlog4j.properties.file=[path-to-log4j.properties]
```

where [path-to-log4j.properties] is the path to the log4j.properties file.

3.2.4 Launching myLEAD Server

Go to <TOMCAT_HOME>/bin directory. If the server is already running, you need to stop it by issuing the following command:

```
./shutdown.sh
```

Start the Tomcat server by doing the following:

```
./startup.sh
```

Open up a web browser and type in the following URL:

```
http://[host]:[port]/ogsa/services/ogsadai/MyLeadGDSF
```

where [host] is the host name of the myLEAD server, and [port] is the port number used by the Tomcat server. This should bring up a WSDL document.

3.3 Installation of the myLEAD Client Service

3.3.1 Setting Environment Variables

The myLEAD client requires two environment variables to be set:

- `MYLEAD_HOME`
- `CLASSPATH`

Please see section 3.2.3 for details on how to set them.

3.3.2 Copying Required Files

The myLEAD client requires the following jars to be pointed to by the CLASSPATH environment variables:

- `axis.jar`
- `cog-axis.jar`
- `cog-jglobus.jar`
- `commons-discovery.jar`
- `commons-logging.jar`
- `jakarta-oro-2.0.8.jar`
- `jaxrpc.jar`
- `log4j-1.2.8.jar`
- `lucene-1.4.3.jar`
- `ogsadai-activities.jar`
- `ogsadai-activities-Indiana2.jar`
- `ogsadai-core.jar`
- `ogsadai-startup-tomcat.jar`
- `ogsadai-ogsi.jar`
- `ogsa.jar`
- `saaj.jar`
- `wsdl4j.jar`
- `xalan.jar`
- `xmldb.jar`
- `xercesImpl.jar`
- `xmlParserAPIs.jar`
- `xmlsec.jar`
- `xpp3-1.1.3.4.B.jar`
- `leaddai.jar`

In addition, place log4j.properties somewhere and set the environment variable MYLEAD_HOME to that directory.

Note: Above files are available from the myleadClientService1.2.tar from the mylead distribution site.

3.4 Installation of the myLEAD Agent Service

The following abbreviations are used:

- <JAVA_HOME>
- <ANT_HOME>
- <MYLEAD_HOME>

3.4.1 Configuration of the MyLead Agent Service

MyLead Agent Service accesses mylead server, and notification broker. The information regarding remote services is configured in mylead.properties.
mylead.properties contains,

- mylead.serverURL: location of the mylead server. You can set a URI of the OGSA-DAI service group registry.
- mylead.notificationBrokerURL: location of the WS notification/eventing broker.
- mylead.agentURL: location of current mylead agent.
- mylead.agentNotifListenerPort: port number that mylead agent consumes the notification.
- mylead.notifconsumerURL: location of the mylead agent for notification broker. This URL should include the port number for consuming the notification. Please note that this is not a agent service URL for service requester.
- mylead.adminDN: mylead requires administrator's user id, also known as administrator's DN. This is used for administrative activities for example creating new user account or modifying it. Also this adminDN must be valid.
- Mylead.sizeofconnectionpool: the number of connections between agent service and mylead server. Default value is 10. All of mylead requests share connections which are initiated when mylead agent service is launched.

For example,

```
mylead.serverURL=http://myleadserver.mylead.edu:10081/ogsa/services/ogs
adai/DAIServiceGroupRegistry
mylead.notificationBrokerURL=notifserver.test.edu:12346
mylead.agentURL=agent.mylead.edu:10081
mylead.agentNotifListenerPort=19999
mylead.notifconsumerURL=agent.mylead.edu:19999
mylead.adminDN=myleadadmin
mylead.sizeofconnectionpool=10
```

If you don't want to process any notification, please configure mylead.agentNotifListenerPort as 0.

Please note that there should NOT be a space within a line of configuration.

3.4.2 Run the MyLead Client Service

In the mylead agent directory, type,

```
./run.sh mylead –handle  [mylead service URL] –port  [mylead service
port]
```

For example,

```
./run.sh mylead –handle http://test.mylead.edu/myleadclient –port 12345
```

3.4.3 Compile MyLead Agent Service

MyLead is an open source project. If the developers require to compile mylead agent service,

```
ant prepare
ant
```

4 Testing the Installation

The myLEAD v1.2 provides testing suite. Testing suite is included in mylead agent package and you can download standalone version as well. If you try standalone version, please configure mylead.properties first. You can follow same instruction in 3.3.1.

To run mylead installation testing suite,

```
./run.sh installtest
```

If you want to test the installation from the 3rd party machine, please download myleadInstallTest1.2.tar from mylead distribution site.

It will test connections to,

- Mylead agent
- Mylead server
- WS Notification/Eventing broker