



# MyLEAD Release V1.3 Installation Guide

Project Title: MyLead

Document Title: myLEAD Release V1.3 Installation Guide

Organization: Indiana University, Distributed Data Everywhere (DDE) Lab

Date: August 14, 2007

Contact: Scott Jensen ([scjensen@cs.indiana.edu](mailto:scjensen@cs.indiana.edu))

Authorship: Scott Jensen, Yiming Sun, and Sangmi Lee Pallickara, Beth Plale

---

1	Introduction.....	2
1.1	myLEAD License .....	2
2	System Specification .....	3
2.1	Prerequisite Products .....	3
2.2	Prerequisite Libraries.....	3
3	Installation Instructions .....	3
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.....	4
3.1.4	Installing OGSA-DAI v6.0 OGSII.....	4
3.2	myLEAD Server .....	6
3.2.1	Setting Environment Variables.....	6
3.2.2	Setting the PATH environment variable to include the Java and ANT executables .....	6
3.2.3	Installing MyLEAD Server .....	6
3.2.3.1	Creating the Database.....	7
3.2.3.2	Deploying the myLEAD Factory in OGSA-DAI .....	8
3.2.3.3	Install the Stored Procedures .....	8
3.2.3.4	Installing the LEAD Schema and myLEAD Activity Schemas .....	8
3.2.3.5	Installing the Jar Files .....	11
3.2.4	Launching myLEAD Server .....	12
3.3	Installation of the myLEAD Client Service .....	13
3.3.1	Setting Environment Variables.....	13
3.3.2	Copying Required Files.....	13

## **1 Introduction**

This document provides an overview of myLEAD system requirements and how to install the myLEAD client and server. The document covers:

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

### **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:

- Red Hat Enterprise Linux 4

### 2.1 Prerequisite Products

myLEAD utilizes some third-party software. The myLEAD system requires the proper installation and configuration of the following 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 OGS

Please see <http://www.ogsadai.org.uk/downloads/archive> for the download of the OGS version compatible with Globus Toolkit 3.2.1. You will need to create an OGSA-DAI account to download software, but setting up an account is free.

- Globus Toolkit WS-Core v3.2.1

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

- Java

JDK version 1.5. Please see <http://java.sun.com>.

- ANT Build tool

To compile myLEAD source distribution using the included ant build file, you will 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 the software.

### 2.2 Prerequisite Libraries

In addition to the above prerequisite products, myLEAD server requires that some additional LEAD and third-party libraries be included in the lib directory of the OGSA-DAI installation in the Tomcat server. All of these jar files are included in the myLEADLibraries1.3.tar.gz tarball – see section 3.2.3.5 for details. Some of these jars must also be included on your classpath when using the myLEAD client. See section 3.3 for details on installing the myLEAD client.

## 3 Installation Instructions

This section covers the process of installing myLEAD server and client version 1.3.8.

## 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 available online at <http://www.mysql.com>.

**Note:** on certain flavors of the Linux platform, client hostnames are not resolved correctly, which may prevent a client from connecting to the database using JDBC. If this occurs, the usual error message is:

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

If this occurs, please connect to MySQL as root and perform the following step:

```
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 that you increase the limit on the maximum number of file descriptors allowed for the user running MySQL. To increase the limit, edit the following file:

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

Add the following two lines to increase the limit 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 that is available online at: <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 instructions on the installation of Tomcat Server.

### 3.1.4 Installing OGSA-DAI v6.0 OGSF

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

```
ogsadai/MyLeadGDSF
```

This name is referenced later in this document for the Service Factory URL. If you used a different name for the factory when installing OGSA-DAI, please use that name in any subsequent reference to the factory in this guide..

**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 you are using the latest XML security jar file instead of the xmlsec.jar provided with the OGSA-DAI package, you may encounter an exception that looks like the following

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

If you encounter this error, 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 was 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 "/usr/local/jdk1.5"
```

For other users, type:

```
export JAVA_HOME "/usr/local/jdk1.5"
```

### 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-1.3.8.jar, which is the myLEAD library.
- myLEADDDB1.3.tar.gz, which contains the database schema definition file and additional scripts for loading namelist metadata attribute definitions.
- myLEADSP1.3.tar.gz, which contains the stored procedures used by myLEAD.
- myLEADActivities1.3.tar.gz, which contains the XML schema files for the myLEAD activities added to OGSA-DAI.
- MyLEAD-ActivityMap1.3.txt, which contains changes to the OGSA-DAI activity map.
- myLEADTypes1.3.tar.gz, which contains the LEAD schema files and additional type files used by the myLEAD activity schemas.
- myLEADLibraries1.3.tar.gz which contains additional myLEAD and third-party libraries needed for myLEAD (in addition to those installed with OGSA-DAI and MySQL).
- log4j.properties, which contains the configuration used by log4j.

### 3.2.3.1 Creating the Database

The first step to installing the myLEAD server is to create the database. While the `mylead1.3.txt` script defines 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. To load the table definitions and namelist metadata attribute definitions, exit MySQL, unzip the `myleadDB1.3.tar.gz` tarball, and run the `LoadDatabase.sh` script in that file to load the table and metadata attribute definitions. The script takes the password of the root MySQL user as a parameter.

#### 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 -R -u <user> -p mcs_lead > <file>
```

This will create `<file>` with all of the table definitions, stored procedure 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 Grid Data Service Factory (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 should use a password that is harder to guess – just be sure to enter that password when prompted for the database password during the OGSA-DAI 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 '[com.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'. (**NOTE:** enter the password actually used)

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 Install the Stored Procedures

The stored procedures used by myLEAD are contained in the `myLEADSP1.3.tar.gz` tarball. To install the stored procedures, first unzip the tarball, which creates a directory named `myLEADSP1.3` 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 or paths for your environment. The script assumes that the user is "root" and the password for the root MySQL user is expected as a parameter.

### 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 `myLEADActivities1.3.tar.gz` tarball. Unzipping this file will create a directory named `myLEADActivities1.3` that contains the 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 their respective schema and Java implementation class 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-ActivityMap1.3.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 `myLEADTypes1.3.tar.gz` tarball, which unzips to a directory named `myLEADTypes1.3`. 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 the schema location settings in the following files need to be updated to reflect the port you are using:

- `general_types.xsd`
- `LeadFGDC.xsd`

- LEAD.xsd
- DataQualityInformation.xsd
- EntityandAttributeInformation.xsd
- TimePeriodInformation.xsd

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"/>
```

Each of the other files listed above contains one or more schema imports or includes similar to that shown above for the `general_types.xsd` schema - the port should be updated for each of these. The imports and includes are always at the start of the schema.

### 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 `myLEADTypes1.3.tar.gz` tarball contains all of the original schema plus the 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="http://schemas.leadproject.org/2007/01/lms/lead"
  xmlns:FGDC="http://schemas.leadproject.org/2007/01/lms/fgdc"
  xmlns:LE="http://schemas.leadproject.org/2007/01/lms/leadelements"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
```

### 3.2.3.5 Installing the Jar Files

All of the server code is contained in the myLEAD `leaddai-1.3.8.jar` file. In addition to this jar file there are eight other jar files that must also be installed.

- `MyLeadResponse-1.3.5.4.jar`:  
This jar contains the XML Bean for the myLEAD query response.
- `lead-metadata-1.8.jar`:  
This jar contains the XML Bean for the LEAD Metadata Schema (LMS).
- `lead-metadata-util-1.6.jar`:  
This jar contains a utility for generating an XML Bean with the minimal metadata required to create a LMS valid document.
- `ogsadai-activities-Indiana2.jar`:  
The second is a jar provided by the OGSA-DAI team that contains a synchronous version of the OGSA-DAI deliver-to-stream activity.
- `xpp3-1.1.3.4.B.jar`:  
This jar file contains the XPP3 Pull Parser.
- `mysql-connector-java-3.1.11-bin.jar`:  
This jar file is the MySQL Connector/J installed when installing MySQL.
- `jsr173_1.0_api.jar` and `xbean.jar`:  
These jar files are available from the XML Beans installation and are needed to use the XML Beans for the LMS and the myLEAD query response.

All of these files are included in the `myLEADLibraries1.3.tar.gz` tarball and should be copied to the following directory: The `log4j-1.2.8.jar` library should already be installed as part of the OGSA-DAI installation process, but it is also included in the additional libraries tarball.

```
<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 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 3.2.1 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
- jaxrpc.jar
- jsr173\_1.0\_api.jar
- leaddai-1.3.8.jar
- lead-metadata-1.8.jar
- log4j-1.2.8.jar
- MyLeadResponse-1.3.5.4.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
- xbean.jar
- xmldb.jar
- xercesImpl.jar
- xmlParserAPIs.jar
- xmlsec.jar
- xpp3-1.1.4c.jar

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

Note: Above files are available from the myLEADClient1.3.tar.gz from the myLEAD distribution site.