MyLEAD Client Service Release V1.3.8
Installation and Developer’s Guide
(a.k.a. “myLEAD agent”)
Introduction ................................................................................................................. 4
1.1 MyLEAD License ................................................................................................ 4
 System Specification ................................................................................................... 5
 2.1 Prerequisite Products ........................................................................................ 5
 2.2 Required Libraries ............................................................................................ 5
 Installing the MyLEAD Agent Service....................................................................... 7
 3.1 Installation ............................................................................................................ 7
 3.2 Setting Up Agent’s Database ............................................................................... 8
 Configuration and Execution of the Agent ................................................................. 9
 4.1 Configuration of the Agent’s Properties File ....................................................... 9
 4.1.1 URLs of other Core Services ........................................................................ 9
 4.1.2 Properties for Notification System .................................................................. 9
 4.1.3 Properties for Database ............................................................................... 10
 4.1.4 File Transfer Topic IDs ............................................................................... 10
 4.1.5 Properties for File Transfer ......................................................................... 10
 4.1.6 Properties for MyLEAD Connection Pool .................................................. 11
 4.1.7 Properties for Namelist File Processing ...................................................... 11
 4.1.8 Miscellaneous Properties ............................................................................ 12
 4.2 Configuration of Shell Script ............................................................................. 12
 4.3 Execution of the Agent ....................................................................................... 12
 Access to the MyLEAD Agent Service .................................................................... 13
 MyLEAD Agent APIs ............................................................................................... 14
 6.1 Working with XML Beans ................................................................................. 14
 6.2 Operational APIs ............................................................................................... 14
 6.2.1 Creating a Project ........................................................................................ 14
 6.2.2 Creating an Experiment .............................................................................. 15
 6.2.3 Creating a Collection .................................................................................. 16
 6.2.4 Registering a File ........................................................................................ 16
 6.2.5 Registering Multiple Files ........................................................................... 17
 6.2.6 Adding Attributes ......................................................................................... 18
 6.2.7 Updating Attributes ..................................................................................... 18
 6.2.8 Deleting an Object ........................................................................................ 19
 6.2.9 Moving an Object ....................................................................................... 19
 6.2.10 Creating a New Account ........................................................................... 20
 6.3 Query APIs ......................................................................................................... 21
 6.3.1 Query to MyLEAD ..................................................................................... 21
 6.3.2 Context Query to MyLEAD ........................................................................ 21
 6.3.3 Retrieving an Entire Sub-tree from Workspace .......................................... 22
 6.3.4 Query by ID ................................................................................................. 22
 6.3.5 Query by Multiple IDs ................................................................................ 22
 6.3.6 Retrieving Workspace Structure ................................................................ 22
 6.3.7 Checking User Account’s Existence ........................................................... 23
 Appendix ................................................................................................................... 24
 7.1 MyLEAD Agent Service WSDL ........................................................................ 24
1 Introduction

The myLEAD Agent service is a service running above the myLEAD server. It actively keeps track of user experiments and workflows, and intelligently manages users’ workspace. The Agent is a stateful service that saves its clients from a heavy dependency stack if they were to interact with the myLEAD server directly. This document provides instructions on the installation and configuration of the myLEAD Agent service, and also serves as a guide to the developers accessing myLEAD server through the myLEAD client service. The document covers:

- How to install and configure the myLEAD Agent
- How to access the myLEAD Agent service from a client
- Description of myLEAD Agent APIs

1.1 MyLEAD License

The file doc/myLEAD-License.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

The myLEAD Agent is designed to work on, and has been tested under, the following platforms:

- Redhat Linux

2.1 Prerequisite Products

Before you start developing a requester to the myLEAD Agent service, the myLEAD system should be installed and running properly. Below is a list of tools/services upon which the Agent depends:

- JDK 1.5 or higher
- ANT 1.6.2 or higher
- MySQL 5.0 or higher
- myLEAD server 1.3.5
- Data Movement and Naming Service (DaMN)

For information about the installation of the myLEAD server system, please refer to its installation guide.

2.2 Required Libraries

Note: You can find all the required library jars in the myLEAD Agent package. Individual jar files reside in subdirectories under the lib directory for better organization:

- axis.jar
- axis2-0.93.jar
- cog-axis.jar
- cog-jglobus.jar
- commons-discovery.jar
- commons-logging.jar
- cryptix32.jar
- cryptix-asn1.jar
- damn_services_types-0.6.jar
- jaxrpc.jar
- jcd-jdk13-131.jar
- jsr173_api.jar
- lead-metadata-1.8.jar
- lead-metadata-util-1.6.1.jar
- leaddai-1.3.5.5.jar
- log4j-1.2.13.jar
- mylead_namelist_attr-1.0.6.jar
- mylead_notification-0.5.2.jar
- MyLeadResponse-1.3.5.4.jar
- mysql-connector-java-3.1.10-bin.jar
- ogsa.jar
- ogsa-activities.jar
- ogsadai-activities-Indiana2.jar
- ogsadai-activities-r6-syncStream.jar
- ogsadai-core-r6-connectionPooling.jar
- ogsadai osgi.jar
- ogsadai-tools.jar
- puretls-0.9b4.jar
- saaj.jar
- servlet.jar
- workflow_tracking-2.3.3.jar
- workflow_tracking_types-2.3.3.jar
- wsdil4j.jar
- wsmg-1.76.7.jar
- xalan.jar
- xbean-2.2.0.jar
- xercesImpl.jar
- xmldb.jar
- xmlParserAPIs.jar
- xmlsec.jar
- xpp3-1.1.3_8.jar
- xpp3_xpath-1.1.3_8.jar
- xsul-2.10.4.jar
- xsul_xwsdlc-2.10.4.jar
- xutil-0.2.jar
3 Installing the MyLEAD Agent Service

3.1 Installation

The installation of the myLEAD Agent Service is easy – just unzip the distribution package to a directory. This document will refer to the installation directory as $AgentPath.

After the package is successfully unzipped, it should contain at least the following files and directories. Please check to make sure they are all present for the successful operation of the Agent service:

```
$AgentPath> ls
build.xml
classpath.sh
config/
etc/
lib/
LICENSE.txt
log4j.properties
myleadagent.properties
run.sh
src/
```

File attributes may or may not be retained by the distribution package, and therefore, it is recommended to do the following commands to make the scripts executable under Linux:

```
$AgentPath> dos2unix *.sh
$AgentPath> chmod 755 *.sh
```

The next step is to compile the Agent’s typelib, assuming Java and Ant are both installed:

```
$AgentPath> ant generate
```

After this step, there should be a new directory called “generated”

```
$AgentPath> ls
... 
generated/
...
```

Now it is time to compile the Agent’s service code:

```
$AgentPath> ant
```

If the compilation is successful, there is yet another new directory called “build”

```
$AgentPath> ls
build/
... 
```
3.2 Setting Up Agent’s Database

In the directory etc/, there is a database SQL script that can be used to create the database for the Agent, and it is assumed the mySQL database has already been installed. The following steps guide the database setup:

```bash
$AgentPath> cd etc
$AgentPath/etc> ls
  myleadagent_dbschema.sql
```

To create the database, do the following commands:

```bash
$AgentPath> mysql -u root
mysql> CREATE DATABASE agent_db;
mysql> GRANT ALL ON agent_db.* TO 'myleadagent'@'localhost' IDENTIFIED BY 'agentpassword';
mysql> GRANT ALL ON agent_db.* TO 'myleadagent'@'%' IDENTIFIED BY 'agentpassword';
```

Here, we assume the name of the database is `agent_db`, the username and password for logging in are `myleadagent` and `agentpassword`, respectively.

Edit the first line of the file `myleadagent_dbschema.sql` so that the database name is consistent with what was just created:

```sql
USE agent_db;
```

After editing the file, do the following command to create tables in the database:

```bash
$AgentPath/etc/> mysql -u root < myleadagent_dbschema.sql
```
4 Configuration and Execution of the Agent

Before the Agent service can be launched, there are a few files that must be configured according to the environment in which it was installed.

4.1 Configuration of the Agent’s Properties File

AGENT_PROPERTY=PROPERTY_VALUE
AGENT_PROPERTY_COUNT=n
AGENT_PROPERTY_ARRAY1=PROPERTY_VALUE
...
AGENT_PROPERTY_ARRAYn=PROPERTY_VALUE

The Agent uses many different services, and therefore, it relies on its properties file to configure how it interacts with them. A typical property of the Agent can either be a name-value pair, or an array of name-value pairs. In the case of an array, there is always a property indicating the total number of elements to look for, and the array itself has unpadded index numbers directly appended to the end of the property names, and the index number also starts at 1. Sample formats of properties are shown above. The following sections will explain those properties by their groups in details.

4.1.1 URLs of other Core Services

MYLEAD_SERVER_URL
DAMN_SERVICE_WSDL_LOCATION

The Agent depends on two core services to function, which are myLEAD server, and Data Movement and Naming (DaMN) service. MYLEAD_SERVER_URL points to where myLEAD server resides, and DAMN_SERVICE_WSDL_LOCATION points to the WSDL location of the DaMN service.

4.1.2 Properties for Notification System

MYLEAD_AGENT_NOTIF_PORT
MYLEAD_NOTIFICATION_BROKER_COUNT
MYLEAD_NOTIF_BROKER_URLI
MYLEAD_NOTIF_MESSAGEBOX_URLI
...
MYLEAD_NOTIF_BROKER_URLIn
MYLEAD_NOTIF_MESSAGEBOX_URLIn

The Agent also uses the messenger service and messagebox service to receive notifications from workflow execution, as well as asynchronous file transfers. MYLEAD_AGENT_NOTIF_PORT specifies the port number that the Agent uses to receive the notification messages. MyLEAD Agent supports the use multiple message brokers and messageboxes, and MYLEAD_NOTIFICATION_BROKER_COUNT specifies the number of brokers, and it is assumed that for each broker, there is an associated messagebox. MYLEAD_NOTIF_BROKER_URLIn and MYLEAD_NOTIF_MESSAGEBOX_URLIn are arrays to specify the URLs of the brokers and messageboxes, respectively.
4.1.3 Properties for Database

<table>
<thead>
<tr>
<th>Property Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYLEAD_AGENT_DATABASE_LOCATION</td>
</tr>
<tr>
<td>MYLEAD_AGENT_DATABASE_USERNAME</td>
</tr>
<tr>
<td>MYLEAD_AGENT_DATABASE_PASSWORD</td>
</tr>
</tbody>
</table>

The three properties in this group specify how the Agent accesses its database. 
MYLEAD_AGENT_DATABASE_LOCATION is the URI to the database that was set up in Section 3.2; and 
MYLEAD_AGENT_DATABASE_USERNAME and MYLEAD_AGENT_DATABASE_PASSWORD are to specify the login username and password, respectively.

4.1.4 File Transfer Topic IDs

<table>
<thead>
<tr>
<th>Topic ID Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGISTER_FILE_TRANSFER_TOPIC_ID</td>
</tr>
<tr>
<td>SPECIAL_FILE_TRANSFER_TOPIC_ID</td>
</tr>
<tr>
<td>NOTIFICATION_FILE_TRANSFER_TOPIC_ID</td>
</tr>
</tbody>
</table>

The Agent may transfer three different kinds of files: regular files, namelist files, and notification log files. For each kind of transfer, the Agent makes a separate subscription to the notification broker, each under a different topic. Those three properties, REGISTER_FILE_TRANSFER_TOPIC_ID, SPECIAL_FILE_TRANSFER_TOPIC_ID, and NOTIFICATION_FILE_TRANSFER_TOPIC_ID are to specify the actual topics for the three subscriptions, respectively.

4.1.5 Properties for File Transfer

<table>
<thead>
<tr>
<th>Property Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYLEAD_REPOSITORY_DESTINATION</td>
</tr>
<tr>
<td>MYLEAD_REPOSITORY_DESTINATION_TYPE</td>
</tr>
<tr>
<td>MYLEAD_SPECIAL_FILE_CACHE_NET</td>
</tr>
<tr>
<td>MYLEAD_SPECIAL_FILE_CACHE_NET_TYPE</td>
</tr>
<tr>
<td>MYLEAD_SPECIAL_FILE_CACHE_LOCAL</td>
</tr>
<tr>
<td>MYLEAD_SPECIAL_FILE_CACHE_LOCAL_TYPE</td>
</tr>
<tr>
<td>MYLEAD_NOTIFICATION_LOGDIRECTORY</td>
</tr>
<tr>
<td>MYLEAD_NOTIFICATION_LOGDIRECTORY_TYPE</td>
</tr>
<tr>
<td>MYLEAD_NOTIFICATION_LOGDIRECTORY_NET</td>
</tr>
<tr>
<td>MYLEAD_NOTIFICATION_LOGDIRECTORY_NET_TYPE</td>
</tr>
</tbody>
</table>

To transfer the files, the Agent must tell the DaMN service regarding the destination location. In addition, the Agent also needs file system directories for logging and processing of namelist files. 
MYLEAD_REPOSITORY_DESTINATION specifies where myLEAD repository is located.

MYLEAD_SPECIAL_FILE_CACHE_NET is the net address of a location to which namelist files can be transferred by the DaMN service, this is because DaMN service can only transfer files using Internet protocols such as gridftp.

MYLEAD_SPECIAL_FILE_CACHE_LOCAL is the local file system path to the directory in where namelist files are temporary transferred for processing, and normally it should point to the same physical space defined by MYLEAD_SPECIAL_FILE_CACHE_NET.
MYLEAD_NOTIFICATION_LOG_DIRECTORY is the local file system path to the directory in where workflow tracking notification messages are logged.

MYLEAD_NOTIFICATION_LOG_DIRECTORY_NET is the net address that points to the same space defined by MYLEAD_NOTIFICATION_LOG_DIRECTORY, so that when a workflow finishes, the log file can be transferred to myLEAD repository by the DaMN service.

In addition, each of the properties stated above has an associated _TYPE property, which could have either a “manual” or an “auto” as its value. This is because the DaMN service supports two types of destinations. A “manual” destination type is to allow the user to specify exactly where a file should be transferred to; an “auto” destination is a logical name for a storage location known by the DaMN service, but the user does not need to know exactly where this storage is. An example of an auto destination would be “damn:iu:lead”, which tells the DaMN service to transfer a file to myLEAD-repository without concerning about where the repository is physically located.

Obviously the _TYPE properties are not meaningful for locations that are local paths to file system directories that do not involve the DaMN service, and they are there only for better organization of the properties.

### 4.1.6 Properties for MyLEAD Connection Pool

<table>
<thead>
<tr>
<th>MYLEAD_CONNECTION_POOL_SIZE</th>
</tr>
</thead>
</table>

The MyLEAD Agent keeps a pool of pre-connected connections to the myLEAD server for scalability and performance. MYLEAD_CONNECTION_POOL_SIZE specifies the number of connections in the pool.

### 4.1.7 Properties for Namelist File Processing

<table>
<thead>
<tr>
<th>LEAD_THEME_KT</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPECIAL_FILE_COUNT</td>
</tr>
<tr>
<td>SPECIAL_FILE_SOURCE1</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>SPECIAL_FILE_SOURCEN</td>
</tr>
</tbody>
</table>

The myLEAD Agent uses shims to process different kinds of namelist files, and these properties specify what indicates a namelist file and what types of namelist files it can process. For more details, please refer to the Section 6.2.4 on registering a file.

LEAD_THEME_KT defines the cue string in the <themekt> node of a file’s metadata to signify a namelist file.

SPECIAL_FILE_COUNT defines the number of different types of namelist files the myLEAD service can process, and SPECIAL_FILE_SOURCEN is an array of the prefixes that are present in the <themekey> node of the file’s metadata, as well as the keywords to feed into the shim.

It is not recommended to make arbitrary modifications to the SPECIAL_FILE_SOURCEN prefixes because the shims depend on them to determine the type of namelist files. If an
administrator needs to make changes to these properties, please make sure the shim will also recognize the prefix strings.

4.1.8 Miscellaneous Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOG4J_PATH</td>
<td>Points to the log4j.properties file. The Agent uses log4j to log useful information. For more details on configuration of log4j, please refer to log4j’s manual.</td>
</tr>
<tr>
<td>OLD_ONLINK_THEME_KT</td>
<td>Specifies the value used for &lt;themekt&gt; node to store original URLs of a file. This is a legacy property and will likely be removed in future versions.</td>
</tr>
</tbody>
</table>

4.2 Configuration of Shell Script

```bash
PATH_TO_AGENT=$AgentPath

JAVA_OPTS="-DMYLEAD_AGENT_HOME=$PATH_TO_AGENT
-Dmylead.log4j.file=$PATH_TO_AGENT/log4j.properties
-Dcertskey=some/path/to/hostcertkey.pem
-Dtrustedcerts=some/path/to/trusted_cas.pem

$JAVA_OPTS"

#$CMD

$CMD 2>&1 | nohup some/path/to/cronolog $PATH_TO_AGENT/cronolog/log-%Y-%m-%d-%H.log
```

Three lines in the shell script run.sh are likely to need configuration. PATH_TO_AGENT should point to the installation directory of the Agent. In addition, it is necessary to specify a few Java properties and pass them into the JVM as JAVA_OPTS.

MYLEAD_AGENT_HOME is where the Agent finds its home directory. The property mylead.log4j.file is used by the client jar to myLEAD server to find the log4j.properties file. In addition, it also passes in certskey and trustedcerts for communication using SSL over HTTP.

The shell script also uses cronolog for log rotation. If this installation of the Agent also uses cronolog, edit the last line of the shell script so it points to the property path in which cronolog resides, as well as where the logs should be written to. However, if cronolog is not used, please comment out this line, and replace it with $CMD.

4.3 Execution of the Agent

```bash
$AgentPath> nohup ./run.sh mylead <port_number> &
```

To start the Agent service, simply execute the command above and specify a port number on which the Agent listens for incoming connections.
5  Access to the MyLEAD Agent Service

First, please make sure you have a valid account on the myLEAD server. Also make sure that the myLEAD server, notification broker, messagebox service, and the Data Movement and Naming service are accessible.

To access the myLEAD Agent service, service requesters should include:

```java
import xsul.XsulVersion;
import xsul.invoker.puretls.PuretlsInvoker;
import xsul.wsdl.WsdlResolver;
import xsul.wsif.spi.WSIFProviderManager;
import xsul.wsif_xsul_soap_gsi.Provider;
import xsul.xwsif_runtime.WSIFClient;
import xsul.xwsif_runtime.XmlBeansWSIFRuntime;
import edu.indiana.dde.mylead.agent.MyLEADAgent_PortType;
```

Since most services are running using SSL for security, it is necessary to provide the Agent client with a certificate file and a key file in order to contact the Agent service. The paths to the files are usually passed into the client as Java properties:

```java
String trustedCerts = System.getProperty("trustedcerts");
String certsKey = System.getProperty("certskey");

if (wsdlLocation.startsWith("https")) {
    PuretlsInvoker invoker =
        new PuretlsInvoker(certsKey, ",", trustedCerts);

    Provider secureProvider = new Provider(invoker);
    WSIFProviderManager.getInstance().addProvider(secureProvider);
    WsdlResolver.getInstance().setSecureInvoker(invoker);
}
```

The myLEAD Agent service is invoked by generating a client-side stub to the service. For example,

```java
MyLEADAgent_PortType myLEADAgentStub =
    (MyLEADAgent_PortType)XmlBeansWSIFRuntime.newClient(wsdlLocation).
    generateDynamicStub(MyLEADAgent_PortType.class);
```
6 MyLEAD Agent APIs

The myLEAD Agent service v1.3.8 provides several APIs which can roughly be categorized into the following:

- Operational APIs
- Query APIs

Operational APIs are methods that could potentially modify the workspace in myLEAD, and query APIs are read-only calls to myLEAD.

All of the Agent’s method invocations are done via web service. The following sections will provide a brief overview of these methods. Please refer to the Agent’s WSDL file listed in Appendix 7.1 for exact definitions. This file is also included in the distribution package and is located in the `config/` directory.

6.1 Working with XML Beans

All inputs and outputs of a method are in the form of XML beans generated from the WSDL. While only XML Documents (e.g. `CreateProjectInputParamsDocument`) appear in the API signatures, a Document is merely a wrapper; the actual elements are always contained in the root element (e.g. `CreateProjectInputParamsType`) of that Document.

Consequently, throughout the following subsections, while the listed method signatures contain Documents, the detailed explanations on the methods will start with the root element. This discrepancy is intentional to avoid unnecessary complications.

6.2 Operational APIs

The MyLEAD Agent service monitors and assists the management of activities related to creating new collections and registering files. The requests are processed in the sequence of:

```
Creating an active experiment
→ Setting up the active experiment
→ Creating the collection and register files
→ Workflow finished or closed/shutdown the active experiment
```

The above sequence is simplified for developers. For more information about the finite state machine, please refer our publication\(^1\).

6.2.1 Creating a Project

```java
public CreateObjectOutputParamsDocument
createProject(CreateProjectInputParamsDocument input)
```

---

Project is the highest element in the user’s personal workspace. MyLEAD provides flexible organization to the users and services. However, there are very simple requirements to build a structure such as a project and experiment.

The input to this method is a CreateProjectInputParamsType object that contains the following elements:

- a valid user DN
- LEAD metadata for the project
- a boolean flag indicating whether to assign a new resource ID

The LEAD metadata is passed to the myLEAD client service as a String object. It must be a valid according to LEAD metadata v1.8, otherwise myLEAD server will not process the data and will throw an exception.

MyLEAD client service assigns a new resource ID to the LEAD metadata if the parameter assignNewResourceId is set as true. If the creation of the project is successful, myLEAD returns the resourceId of the new object as a String. This resourceId can be used to create child elements of this project.

A project may contain nested projects, experiments, collections, and files. Users or services can combine those elements to build their own structure.

This method returns a CreateObjectOutputParamsType object that contains one of the following elements:

- resource ID of the new project
- fault message

If the operation was successful, the resource ID of the newly created project is returned; otherwise, the fault message contains information on the exception.

### 6.2.2 Creating an Experiment

```java
public CreateObjectOutputParamsDocument createExperiment(CreateExperimentInputParamsDocument input)
```

An experiment is a container representing each experiment in myLEAD. The input to this method is a CreateExperimentInputParamsType object that contains the following elements:

- a valid user DN
- LEAD metadata for the experiment
- resource ID of its parent (usually a project’s ID)
- a boolean flag indicating whether to assign a new resource ID

The parameters are similar to those for creating projects, with the exception of an extra parameter that indicates what the parent is.

An experiment may contain nested experiments, collections, and files, but it cannot contain projects.
This method returns a `CreateObjectOutputParamsType` object. Please see Section 6.2.1 for details on this type of objects.

### 6.2.3 Creating a Collection

```
public CreateObjectOutputParamsDocument
    createCollection(CreateCollectionInputParamsDocument input)
```

Collections provide flexible structure in the myLEAD personal metadata catalog. Users can build a collection under a project, an experiment, and even another collection.

The input to this method is a `CreateCollectionInputParamsType` object that contains the following elements:

- a valid user DN,
- LEAD metadata for the collection
- resource ID of its parent
- a boolean flag indicating whether to assign a new resource ID

A collection can contain nested collections and files as its children, but it cannot contain projects or experiments.

This method returns a `CreateObjectOutputParamsType` object. Please see Section 6.2.1 for details on this type of objects.

### 6.2.4 Registering a File

```
public CreateObjectOutputParamsDocument
    registerFile(RegisterFileInputParamsDocument input)

public CreateObjectOutputParamsDocument
    registerFileMetadata(RegisterFileMetadataInputParamsDocument input)
```

There are two ways to register a file into a user’s workspace. The first method is to register the physical file. The Agent will contact the Data Movement and Naming service to transfer the file from its source location to the myLEAD repository where files are being safely kept. When this is successful, the Agent stores the metadata for the file into the user’s workspace.

The input to both methods is a `RegisterFileInputParamsType` object that contains the following elements:

- a valid user DN
- LEAD metadata for the file
- source URL of the file
- resource ID of the parent
- a boolean flag indicating whether to assign a new resource ID

Since a file can be potentially large and may take a long time for the transfer to complete, the registering of a file uses asynchronous mode so a user may continue with work after made the request to register a file. A successful return from this method call is merely an indication that the file transfer request has been received.
This asynchronous mode of operation applies to any work that involves file transfer. A few paragraphs later in this guide, it will explain the processing of namelist files, and it also uses the asynchronous mode.

The second method is to directly store the file’s metadata without physically transferring it into the myLEAD repository. This method is not recommended for use unless there is a very strong reason for doing so. For one thing, a file registered using the second method is not visible to the Data Movement and Naming service and therefore it cannot be later resolved from this service. The parameters for calling this method is identical to the first method, and it is strongly recommended to provide a source URL of the file even though it is not being used by the Agent for data transfer. In most cases, this method call is synchronous, i.e. the metadata is stored in the myLEAD server before returning to the client. The only exception is if the file is a namelist file, which is explained below.

A namelist file is a special kind of file that contains configuration parameters for a user’s experiment. Those files can be specially processed so that the parameters are extracted and stored as attributes to the parent of the file. A user who wishes to process a namelist file must make sure the LEAD metadata for the file contains a `<themekt>` node whose value is NAMELIST or any string that is configured by the Agent’s administrator, and it also must have an associated `<themekey>` node containing one of the predefined prefixes. This list of prefixes is also configurable, and currently the Agent is capable of processing the following:

- Terrain
- WRFSTATIC
- ADASIntrp
- EXT2ARPS
- LATERAL
- ARPS2WRF
- WRFEXTSND
- ADAS-EXT2ARPS
- WRF2ARPS
- ARPSPLT

It is worth noting that the processing of namelist files requires special shim (myLEAD_namelist_attr-1.0.6.jar) which relies on the prefixes to determine the type of a namelist file, and therefore, the Agent service’s administrator must make sure the shim can handle a customized prefix string before making changes to this list.

In order to extract the parameters, the Agent must have access to the namelist files. Consequently the files will be transferred to a temporary space reserved for the Agent and the contents read, and the overall time may take even longer.

Both methods return a `CreateObjectOutputParamsType` object. Please see Section 6.2.1 for details on this type of objects.

### 6.2.5 Registering Multiple Files

```java
generic BulkCreateObjectOutputParamsDocument
bulkRegisterFile(BulkRegisterFileInputParamsDocument input)
generic BulkCreateObjectOutputParamsDocument bulkRegisterFileMetadata
```
The methods for registering a file explained in Section 6.2.4 also have bulk counter-parts. The bulk methods can be used to register multiple files under the same parent, and the input to both methods is a BulkRegisterFileInputParamsType object that contains the following elements:
- a valid user DN
- an array of LEAD metadata of the files
- an array of source URLs of the files
- resource ID of the common parent
- an array of boolean flags indicating whether to assign a new resource ID for each file

Both methods return a BulkCreateObjectOutputParamsType object that contains one of the following elements:
- an array of resource IDs
- fault message

If the operation was successful, the resource IDs of all the files being registered are returned in an array; otherwise, the fault message contains information on the first exception the method encounters.

### 6.2.6 Adding Attributes

public AddAttributesOutputParamsDocument addAttributes(AddAttributesInputParamsDocument input)

The LEAD metadata schema contains a lot of different attributes, and it is unreasonable to assume a user has all at an object’s creation time, and therefore, the myLEAD service provides a way for a user to add new attributes to an existing object at a later time.

The input to this method is an AddAttributeInputParamsType object that contains the following elements:
- a valid user DN
- an existing object’s resource ID
- an array of XML segments containing the attributes to be added

This method returns an AddAttributesOutputParamsType object that contains one of the following elements:
- results
- fault message

If the operation was successful, a string indicating the success is returned; otherwise, the fault message contains information on the exception.

### 6.2.7 Updating Attributes

public UpdateAttributesOutputParamsDocument updateAttributes(UpdateAttributesInputParamsDocument input)
Aside from adding new attributes, a user may also wish to update existing attributes as progress is made. MyLEAD service provides an API for this as well, and the input to this method is an `UpdateAttributesInputParamsType` object that contains the following elements:

- a valid user DN
- an array of `UpdateAttributeHolderType` objects

Each `UpdateAttributeHolderType` object contains information necessary for making updates to an attribute, and its anatomy is as follows:

- an XML segment containing the up-to-date attribute
- resource ID of the object to which the attribute belong
- an optional position in case multiple instances of an attribute exist for the object
- an optional timestamp for version control

This method returns an `UpdateAttributesOutputParamsType` object that contains one of the following elements:

- results
- fault message

If the operation was successful, a string indicating the success is returned; otherwise, the fault message contains information on the exception.

### 6.2.8 Deleting an Object

```java
public DeleteObjectOutputParamsDocument deleteObject(DeleteObjectInputParamsDocument input)
```

This method allows a user to delete an object and all its children he/she no longer needs. The input to this method is a `DeleteObjectInputParamsType` object that contains the following elements:

- a valid user DN
- resource ID of the top level object

This method returns a `DeleteObjectOutputParamsType` object that contains one of the following elements:

- results
- fault message

If the operation was successful, a string indicating the success is returned; otherwise, the fault message contains information on the exception.

Be very careful about this method. It could be dangerous since it deletes an entire subtree.

### 6.2.9 Moving an Object

```java
public MoveObjectOutputParamsDocument moveObject(MoveObjectInputParamsDocument input)
```

This method allows a user to move a subtree to be under a new parent. The input to this method is a `MoveObjectInputParamsType` object that contains the following elements:

- a valid user DN
- resource ID of the top level object
• resource ID of the new parent

This method returns a `MoveObjectOutputParamsType` object that contains one of the following elements:
• results
• fault message

If the operation was successful, a string indicating the success is returned; otherwise, the fault message contains information on the exception.

### 6.2.10 Creating a New Account

```java
public QueryOutputParamsDocument
  addNewUser(AddNewUserInputParamsDocument input)
```

This method is actually of administrative purpose. By calling this method, a new user account can be created. Its input is an `AddNewUserInputParamsType` object that contains the following elements:
• an administrator’s DN
• a `NewUserInfo` object

A `NewUserInfo` object contains information regarding the new user account to be created, and its contents are listed below:
• DN of the new user
• real name of the new user
• an optional element for replica URL
• an optional element for user’s organization
• an optional element for user’s position
• an optional element for user’s address type
• an optional element for user’s street
• an optional element for user’s city
• an optional element for user’s state
• an optional element for user’s zip code
• an optional element for user’s country of residence
• an optional element for user’s phone number
• an optional element for user’s TTD/TTY phone
• an optional element for user’s email address
• an optional element for user’s fax number
• an optional element for user’s homepage URL
• an optional element for user’s hours of contact
• an optional element for user’s institution
• an optional array of `StorageResource` objects

A `StorageResource` object contains information regarding storage locations per user’s preference. It contains the following elements:
• an optional element for resource ID of the storage resource
• an optional element for access URL of the storage resource
• an optional nickname of the storage resource
• an optional array of protocols supported by the storage resource

This method returns a QueryOutputParamsType object that contains one of the following elements:
• results
• fault message

If the operation was successful, a string indicating the success is returned; otherwise, the fault message contains information on the exception.

6.3 Query APIs
MyLEAD Agent v1.3.8 also has a set of APIs for performing queries. These are pass-through APIs that expose functionalities of the myLEAD server. This section only gives a brief overview of these APIs. For more detailed information, please refer to myLEAD server’s documentation.

6.3.1 Query to MyLEAD

```java
public QueryOutputParamsDocument queryLead(QueryLeadInputParamsDocument input)
```

To issue a simple query to the MyLEAD server, a QueryLeadInputParamsType object must be provided as the input, which contains the following elements:
• a valid user DN
• number of results to be returned
• hierarchy filter
• content filter
• target of the query

This method returns a QueryOutputParamsDocument object. Please see Section 6.2.10 for details on this type of objects.

6.3.2 Context Query to MyLEAD

```java
public QueryOutputParamsDocument queryLeadwithContextQuery(
    QueryLeadWithContextQueryInputParamsDocument input)
```

Context query is a powerful feature of MyLEAD service, and it allows a user to issue queries based on an object’s parent and/or children. To issue a context query, a QueryLeadwithContextQueryInputParamsType object must be provided as the input, which contains the following elements:
• a valid user DN
• number of results to be returned
• hierarchy filter
• content filter
• target of the query
• context

This method returns a QueryOutputParamsDocument object. Please see Section 6.2.10 for details on this type of objects.
6.3.3 Retrieving an Entire Sub-tree from Workspace

```java
public QueryOutputParamsDocument
    queryWorkspace(QueryWorkspaceInputParamsDocument input)
```

This method allows a user to retrieve everything in the workspace under a given object. The input to this method is a `QueryWorkspaceInputParamsType` object that contains the following elements:

- resource ID of the top level object
- number of results to be returned

This method returns a `QueryOutputParamsDocument` object. Please see Section 6.2.10 for details on this type of objects.

Depending on the top level object provided, the results from this method may be huge. Use this method with caution.

6.3.4 Query by ID

```java
public QueryOutputParamsDocument
    queryById(QueryByIdInputParamsDocument input)
```

This method is similar to the previous method in that it can query for details of a given object; however, this method supports filtering so a user may selectively choose what to return as the results. The input to this method is a `QueryByIdInputParamsType` object that contains the following elements:

- a valid user DN
- hierarchy filter
- content filter
- resource ID of the top level object

This method returns a `QueryOutputParamsDocument` object. Please see Section 6.2.10 for details on this type of objects.

6.3.5 Query by Multiple IDs

```java
public QueryOutputParamsDocument
    queryByIds(QueryByIdsInputParamsDocument input)
```

There is also a method for querying multiple IDs using the same filters, and its input is a `QueryByIdsInputParamsType` object that contains the following elements:

- a valid user DN
- hierarchy filter
- content filter
- an array of resource IDs

This method returns a `QueryOutputParamsDocument` object. Please see Section 6.2.10 for details on this type of objects.

6.3.6 Retrieving Workspace Structure

```java
public QueryOutputParamsDocument
    queryWorkspaceStructure(
        QueryWorkspaceStructureInputParamsDocument input)
```
This method retrieves a user’s workspace structure by returning only the resource IDs and titles of everything in the user’s workspace without retrieving any other detailed attributes. A typical use of this method is to display the workspace using a tree. The input to this method is a QueryWorkspaceStructureInputParamsType object that contains the following element:
- a valid user DN

This method returns a QueryOutputParamsDocument object. Please see Section 6.2.10 for details on this type of objects.

Combined with the queryById method, they can be used to display a user’s workspace contents with scalability.

### 6.3.7 Checking User Account’s Existence

```java
doesUserHaveMyLeadAccount(SimpleStringParamsDocument input)
```

While every operation or query to myLEAD goes through authorization by the service, it may still be necessary to explicitly check the validity (existence) of a user DN in some circumstances. This method returns a boolean flag to indicate if the specified DN is registered with myLEAD server. The input to this method is a SimpleStringParamsType object that contains the following element:
- a DN in question

This method returns a SimpleBooleanParamsType object that contains one of the following elements:
- a boolean flag
- fault message

If the query was successful, a boolean flag is returned to indicate if the user account exists in myLEAD server; otherwise the fault message contains information on the exception.
7 Appendix

7.1 MyLEAD Agent Service WSDL

```xml
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="myleadagent"
    targetNamespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
    xmlns:typens="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3/xsd"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:soap:encoded="http://schemas.xmlsoap.org/wsdl/soap/encoded/"
    xmlns:soap:encoded:encoded="http://schemas.xmlsoap.org/wsdl/soap/encoded/encoded/"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:xcatPar="http://www.dde.indiana.edu/namespaces/2004/10/xwf/xcat/parameter"
    xmlns:wsdl:encoded="http://schemas.xmlsoap.org/wsdl/encoded/"
    xmlns:wsdl:encoded:encoded="http://schemas.xmlsoap.org/wsdl/encoded/encoded/"
>
<message name="CreateObject_ResponseMessage">
    <part name="CreateObject_Run_OutputParameters" element="typens:CreateObject_OutputParams"/>
</message>
<message name="BulkCreateObject_ResponseMessage">
    <part name="BulkCreateObject_Run_OutputParameters" element="typens:BulkCreateObject_OutputParams"/>
</message>
<message name="CreateProject_RequestMessage">
    <part name="CreateProject_Run_InputParameters" element="typens:CreateProject_InputParams"/>
</message>
<message name="CreateExperiment_RequestMessage">
    <part name="CreateExperiment_Run_InputParameters" element="typens:CreateExperiment_InputParams"/>
</message>
<message name="CreateCollection_RequestMessage">
    <part name="CreateCollection_Run_InputParameters" element="typens:CreateCollection_InputParams"/>
</message>
<message name="RegisterFile_RequestMessage">
    <part name="RegisterFile_Run_InputParameters" element="typens:RegisterFile_InputParams"/>
</message>
<message name="RegisterFileMetadata_RequestMessage">
    <part name="RegisterFileMetadata_Run_InputParameters" element="typens:RegisterFileMetadata_InputParams"/>
</message>
<message name="BulkRegisterFile_RequestMessage">
    <part name="BulkRegisterFile_Run_InputParameters" element="typens: BulkRegisterFile_InputParams"/>
</message>
<message name="BulkRegisterFileMetadata_RequestMessage">
    <part name="BulkRegisterFileMetadata_Run_InputParameters" element="typens:BulkRegisterFileMetadata_InputParams"/>
</message>
<message name="DoesUserHaveMyLeadAccount_RequestMessage">
    <part name="DoesUserHaveMyLeadAccount_Run_InputParameters" element="typens:SimpleString_Params"/>
</message>
<message name="DoesUserHaveMyLeadAccount_ResponseMessage">
    <part name="DoesUserHaveMyLeadAccount_Run_OutputParameters" element="typens:SimpleBoolean_Params"/>
</message>
<message name="CloseMyLeadConnections_RequestMessage">
    <part name="CloseMyLeadConnections_Run_InputParameters" element="typens:SimpleBoolean_Params"/>
</message>
<message name="CloseMyLeadConnections_ResponseMessage">
    <part name="CloseMyLeadConnections_Run_OutputParameters" element="typens:SimpleBoolean_Params"/>
</message>
</definitions>
```
<message name="QueryLeadWithContextQuery_RequestMessage">
  <part name="QueryLeadWithContextQuery_Run_InputParameters" element="typens:QueryLeadWithContextQuery_InputParams"/>
</message>

<message name="QueryLead_RequestMessage">
  <part name="QueryLead_Run_InputParameters" element="typens:QueryLead_InputParams"/>
</message>

<message name="QueryById_RequestMessage">
  <part name="QueryById_Run_InputParameters" element="typens:QueryById_InputParams"/>
</message>

<message name="QueryByIds_RequestMessage">
  <part name="QueryByIds_Run_InputParameters" element="typens:QueryByIds_InputParams"/>
</message>

<message name="QueryWorkspace_RequestMessage">
  <part name="QueryWorkspace_Run_InputParameters" element="typens:QueryWorkspace_InputParams"/>
</message>

<message name="QueryWorkspaceStructure_RequestMessage">
  <part name="QueryWorkspaceStructure_Run_InputParameters" element="typens:QueryWorkspaceStructure_InputParams"/>
</message>

<message name="Query_ResponseMessage">
  <part name="Query_Run_OutputParameters" element="typens:Query_OutputParams"/>
</message>

<message name="AddNewUser_RequestMessage">
  <part name="AddNewUser_Run_InputParameters" element="typens:AddNewUser_InputParams"/>
</message>

<message name="DeleteObject_RequestMessage">
  <part name="DeleteObject_Run_InputParameters" element="typens:DeleteObject_InputParams"/>
</message>

<message name="DeleteObject_ResponseMessage">
  <part name="DeleteObject_Run_OutputParameters" element="typens:DeleteObject_OutputParams"/>
</message>

<message name="MoveObject_RequestMessage">
  <part name="MoveObject_Run_InputParameters" element="typens:MoveObject_InputParams"/>
</message>

<message name="MoveObject_ResponseMessage">
  <part name="MoveObject_Run_OutputParameters" element="typens:MoveObject_OutputParams"/>
</message>

<message name="PublishObjectNotification_RequestMessage">
  <part name="PublishObjectNotification_Run_InputParameters" element="typens:PublishObjectNotification_InputParams"/>
</message>

<message name="PublishObjectNotification_ResponseMessage">
  <part name="PublishObjectNotification_Run_OutputParameters" element="typens:PublishObjectNotification_OutputParams"/>
</message>

<message name="UpdateAttributes_RequestMessage">
  <part name="UpdateAttributes_Run_InputParameters" element="typens:UpdateAttributes_InputParams"/>
</message>

<message name="UpdateAttributes_ResponseMessage">
  <part name="UpdateAttributes_Run_OutputParameters" element="typens:UpdateAttributes_OutputParams"/>
</message>

<message name="AddAttributes_RequestMessage">
  <part name="AddAttributes_Run_InputParameters" element="typens:AddAttributes_InputParams"/>
</message>

<message name="AddAttributes_ResponseMessage">
  <part name="AddAttributes_Run_OutputParameters" element="typens:AddAttributes_OutputParams"/>
</message>

<portType name="MyLEADAgent_PortType">
  <operation name="createProject">
    <input name="CreateProject_RequestMessage" message="wsdlns:CreateProject_RequestMessage"/>
    <output name="CreateProject_ResponseMessage"/>
<operation name="createExperiment">
  <input name="CreateExperiment_RequestMessage" message="wsdlns:CreateExperiment_RequestMessage"/>
  <output name="CreateExperiment_ResponseMessage" message="wsdlns:CreateObject_ResponseMessage"/>
</operation>

<operation name="createCollection">
  <input name="CreateCollection_RequestMessage" message="wsdlns:CreateCollection_RequestMessage"/>
  <output name="CreateCollection_ResponseMessage" message="wsdlns:CreateObject_ResponseMessage"/>
</operation>

<operation name="registerFile">
  <input name="RegisterFile_RequestMessage" message="wsdlns:RegisterFile_RequestMessage"/>
  <output name="RegisterFile_ResponseMessage" message="wsdlns:CreateObject_ResponseMessage"/>
</operation>

<operation name="registerFileMetadata">
  <input name="RegisterFileMetadata_RequestMessage" message="wsdlns:RegisterFileMetadata_RequestMessage"/>
  <output name="RegisterFile_ResponseMessage" message="wsdlns:CreateObject_ResponseMessage"/>
</operation>

<operation name="bulkRegisterFile">
  <input name="BulkRegisterFile_RequestMessage" message="wsdlns:BulkRegisterFile_RequestMessage"/>
  <output name="BulkRegisterFile_ResponseMessage" message="wsdlns:BulkCreateObject_ResponseMessage"/>
</operation>

<operation name="bulkRegisterFileMetadata">
  <input name="BulkRegisterFileMetadata_RequestMessage" message="wsdlns:BulkRegisterFileMetadata_RequestMessage"/>
  <output name="BulkRegisterFile_ResponseMessage" message="wsdlns:BulkCreateObject_ResponseMessage"/>
</operation>

<operation name="doesUserHaveMyLeadAccount">
  <input name="DoesUserHaveMyLeadAccount_RequestMessage" message="wsdlns:DoesUserHaveMyLeadAccount_RequestMessage"/>
  <output name="DoesUserHaveMyLeadAccount_ResponseMessage" message="wsdlns:DoesUserHaveMyLeadAccount_ResponseMessage"/>
</operation>

<operation name="closeMyLeadConnections">
  <input name="CloseMyLeadConnections_RequestMessage" message="wsdlns:CloseMyLeadConnections_RequestMessage"/>
  <output name="CloseMyLeadConnections_ResponseMessage" message="wsdlns:CloseMyLeadConnections_ResponseMessage"/>
</operation>

<operation name="queryLeadWithContextQuery">
  <input name="QueryLeadWithContextQuery_RequestMessage" message="wsdlns:QueryLeadWithContextQuery_RequestMessage"/>
  <output name="Query_ResponseMessage" message="wsdlns:Query_ResponseMessage"/>
</operation>

<operation name="queryLead">
  <input name="QueryLead_RequestMessage" message="wsdlns:QueryLead_RequestMessage"/>
  <output name="Query_ResponseMessage" message="wsdlns:Query_ResponseMessage"/>
</operation>

<operation name="queryById">
  <input name="QueryById_RequestMessage" message="wsdlns:QueryById_RequestMessage"/>
  <output name="Query_ResponseMessage" message="wsdlns:Query_ResponseMessage"/>
</operation>

<operation name="queryByIds">
  <input name="QueryByIds_RequestMessage" message="wsdlns:QueryByIds_RequestMessage"/>
  <output name="Query_ResponseMessage" message="wsdlns:Query_ResponseMessage"/>
</operation>

<operation name="queryWorkspace">
  <input name="QueryWorkspace_RequestMessage" message="wsdlns:QueryWorkspace_RequestMessage"/>
  <output name="Query_ResponseMessage" message="wsdlns:Query_ResponseMessage"/>
<operation name="queryWorkspaceStructure">
  <input name="QueryWorkspaceStructure_RequestMessage" message="wadlns:QueryWorkspaceStructure_RequestMessage"/>
  <output name="Query_ResponseMessage" message="wadlns:Query_ResponseMessage"/>
</operation>

<operation name="addNewUser">
  <input name="AddNewUser_RequestMessage" message="wadlns:AddNewUser_RequestMessage"/>
  <output name="Query_ResponseMessage" message="wadlns:Query_ResponseMessage"/>
</operation>

<operation name="deleteObject">
  <input name="DeleteObject_RequestMessage" message="wadlns:DeleteObject_RequestMessage"/>
  <output name="DeleteObject_ResponseMessage" message="wadlns:DeleteObject_ResponseMessage"/>
</operation>

<operation name="moveObject">
  <input name="MoveObject_RequestMessage" message="wadlns:MoveObject_RequestMessage"/>
  <output name="MoveObject_ResponseMessage" message="wadlns:MoveObject_ResponseMessage"/>
</operation>

<operation name="storePublisherNotification">
  <input name="PublishObjectNotification_RequestMessage" message="wadlns:PublishObjectNotification_RequestMessage"/>
  <output name="PublishObjectNotification_ResponseMessage" message="wadlns:PublishObjectNotification_ResponseMessage"/>
</operation>

<operation name="updateAttributes">
  <input name="UpdateAttributes_RequestMessage" message="wadlns:UpdateAttributes_RequestMessage"/>
  <output name="UpdateAttributes_ResponseMessage" message="wadlns:UpdateAttributes_ResponseMessage"/>
</operation>

<operation name="addAttributes">
  <input name="AddAttributes_RequestMessage" message="wadlns:AddAttributes_RequestMessage"/>
  <output name="AddAttributes_ResponseMessage" message="wadlns:AddAttributes_ResponseMessage"/>
</operation>

</portType>

<binding name="MyLEADAgent_SoapBinding" type="wadlns:MyLEADAgent_PortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http">

    <operation name="createProject">
      <soap:operation soapAction=""/>
      <input name="CreateProject_RequestMessage">
        <soap:body use="literal" namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3" required="true"/>
      </input>
      <output name="CreateProject_ResponseMessage">
        <soap:body use="literal" namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3" required="true"/>
      </output>
    </operation>

    <operation name="createExperiment">
      <soap:operation soapAction=""/>
      <input name="CreateExperiment_RequestMessage">
        <soap:body use="literal" namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3" required="true"/>
      </input>
      <output name="CreateExperiment_ResponseMessage">
        <soap:body use="literal" namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3" required="true"/>
      </output>
    </operation>

  </soap:binding>
</binding>
<operation name="createCollection">
  <soap:operation soapAction=""/>
  <input name="CreateCollection_RequestMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
      required="true"/>
  </input>
  <output name="CreateCollection_ResponseMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
      required="true"/>
  </output>
</operation>

<operation name="registerFile">
  <soap:operation soapAction=""/>
  <input name="RegisterFile_RequestMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
      required="true"/>
  </input>
  <output name="RegisterFile_ResponseMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
      required="true"/>
  </output>
</operation>

<operation name="registerFileMetadata">
  <soap:operation soapAction=""/>
  <input name="RegisterFileMetadata_RequestMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
      required="true"/>
  </input>
  <output name="RegisterFileMetadata_ResponseMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
      required="true"/>
  </output>
</operation>

<operation name="bulkRegisterFile">
  <soap:operation soapAction=""/>
  <input name="BulkRegisterFile_RequestMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
      required="true"/>
  </input>
  <output name="BulkCreateObject_ResponseMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
      required="true"/>
  </output>
</operation>

<operation name="bulkRegisterFileMetadata">
  <soap:operation soapAction=""/>
  <input name="BulkRegisterFileMetadata_RequestMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
      required="true"/>
  </input>
  <output name="BulkCreateObject_ResponseMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
      required="true"/>
  </output>
</operation>
<operation name="doesUserHaveMyLeadAccount">
  <soap:operation soapAction=""/>
  <input name="DoesUserHaveMyLeadAccount_RequestMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namesp.../mylead/myleadagent/1.3"
      required="true"/>
  </input>
  <output name="DoesUserHaveMyLeadAccount_ResponseMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namesp.../mylead/myleadagent/1.3"
      required="true"/>
  </output>
</operation>

<operation name="closeMyLeadConnections">
  <soap:operation soapAction=""/>
  <input name="CloseMyLeadConnections_RequestMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namesp.../mylead/myleadagent/1.3"
      required="true"/>
  </input>
  <output name="CloseMyLeadConnections_ResponseMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namesp.../mylead/myleadagent/1.3"
      required="true"/>
  </output>
</operation>

<operation name="queryLeadwithContextQuery">
  <soap:operation soapAction=""/>
  <input name="QueryLeadWithContextQuery_RequestMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namesp.../mylead/myleadagent/1.3"
      required="true"/>
  </input>
  <output name="Query_ResponseMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namesp.../mylead/myleadagent/1.3"
      required="true"/>
  </output>
</operation>

<operation name="queryLead">
  <soap:operation soapAction=""/>
  <input name="QueryLead_RequestMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namesp.../mylead/myleadagent/1.3"
      required="true"/>
  </input>
  <output name="Query_ResponseMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namesp.../mylead/myleadagent/1.3"
      required="true"/>
  </output>
</operation>

<operation name="queryById">
  <soap:operation soapAction=""/>
  <input name="QueryById_RequestMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namesp.../mylead/myleadagent/1.3"
      required="true"/>
  </input>
  <output name="Query_ResponseMessage">
    <soap:body use="literal"
      namespace="http://www.cs.indiana.edu/dde/namesp.../mylead/myleadagent/1.3"
      required="true"/>
  </output>
</operation>
<soap:operation soapAction=""/>
<input name="QueryByIds_RequestMessage">
<soap:body use="literal"
namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</input>
<output name="Query_ResponseMessage">
<soap:body use="literal"
namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</output>
</operation>

<operation name="queryWorkspace">
<soap:operation soapAction=""/>
<input name="QueryWorkspace_RequestMessage">
<soap:body use="literal"
namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</input>
<output name="Query_ResponseMessage">
<soap:body use="literal"
namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</output>
</operation>

<operation name="queryWorkspaceStructure">
<soap:operation soapAction=""/>
<input name="QueryWorkspaceStructure_RequestMessage">
<soap:body use="literal"
namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</input>
<output name="Query_ResponseMessage">
<soap:body use="literal"
namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</output>
</operation>

<operation name="addNewUser">
<soap:operation soapAction=""/>
<input name="AddNewUser_RequestMessage">
<soap:body use="literal"
namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</input>
<output name="Query_ResponseMessage">
<soap:body use="literal"
namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</output>
</operation>

<operation name="deleteObject">
<soap:operation soapAction=""/>
<input name="DeleteObject_RequestMessage">
<soap:body use="literal"
namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</input>
<output name="DeleteObject_ResponseMessage">
<soap:body use="literal"
namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</output>
</operation>
<operation name="moveObject">
<soap:operation soapAction=""/>
<input name="MoveObject_RequestMessage">
<soap:body use="literal"
namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</input>
<output name="MoveObject_ResponseMessage">
<soap:body use="literal"
namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</output>
</operation>

<operation name="storePublisherNotification">
<soap:operation soapAction=""/>
<input name="PublishObjectNotification_RequestMessage">
<soap:body use="literal"
namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</input>
<output name="PublishObjectNotification_ResponseMessage">
<soap:body use="literal"
namespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</output>
</operation>

<operation name="updateAttributes">
<soap:operation soapAction=""/>
<input name="UpdateAttributes_RequestMessage">
<soap:body use="literal"
namespace="https://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</input>
<output name="UpdateAttributes_ResponseMessage">
<soap:body use="literal"
namespace="https://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</output>
</operation>

<operation name="addAttributes">
<soap:operation soapAction=""/>
<input name="AddAttributes_RequestMessage">
<soap:body use="literal"
namespace="https://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</input>
<output name="AddAttributes_ResponseMessage">
<soap:body use="literal"
namespace="https://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3"
required="true"/>
</output>
</operation>
</binding>

<service name="MyLEADAgent">
<port name="MyLEADAgent_Port" binding="wsdl:MyLEADAgent_SoapBinding">
<soap:address location="http://host.port/location"/>
</port>
</service>

types>
<schema elementFormDefault="unqualified"
targetNamespace="http://www.cs.indiana.edu/dde/namespace/mylead/myleadagent/1.3/xsd"
xmlns="http://www.w3.org/2001/XMLSchema">
<annotation>
  <documentation xml:lang="en">
    This is the WSDL for myLEAD Agent v1.3.8.6
  </documentation>
</annotation>

<element name="CreateProject_InputParams" type="typens:CreateProject_InputParamsType"/>
<element name="CreateObject_OutputParams" type="typens:CreateObject_OutputParamsType"/>
<element name="BulkCreateObject_OutputParams" type="typens:CreateObject_OutputParamsType"/>
<element name="CreateExperiment_InputParams" type="typens:CreateExperiment_InputParamsType"/>
<element name="CreateCollection_InputParams" type="typens:CreateCollection_InputParamsType"/>
<element name="RegisterFile_InputParams" type="typens:RegisterFile_InputParamsType"/>
<element name="RegisterFileMetadata_InputParams" type="typens:RegisterFile_InputParamsType"/>
<element name="BulkRegisterFile_InputParams" type="typens:BulkRegisterFile_InputParamsType"/>
<element name="BulkRegisterFileMetadata_InputParams" type="typens:BulkRegisterFileMetaData_InputParamsType"/>
<element name="SimpleString_Params" type="typens:SimpleString_ParamsType"/>
<element name="SimpleBoolean_Params" type="typens:SimpleBoolean_ParamsType"/>
<element name="QueryLeadWithContextQuery_InputParams" type="typens:QueryLeadWithContextQuery_InputParamsType"/>
<element name="Query_OutputParams" type="typens:Query_OutputParamsType"/>
<element name="QueryLead_InputParams" type="typens:QueryLead_InputParamsType"/>
<element name="QueryById_InputParams" type="typens:QueryById_InputParamsType"/>
<element name="QueryByIds_InputParams" type="typens:QueryByIds_InputParamsType"/>
<element name="QueryWorkspace_InputParams" type="typens:QueryWorkspace_InputParamsType"/>
<element name="QueryWorkspaceStructure_InputParams" type="typens:QueryWorkspaceStructure_InputParamsType"/>
<element name="AddNewUser_InputParams" type="typens:AddNewUser_InputParamsType"/>
<element name="DeleteObject_InputParams" type="typens:DeleteObject_InputParamsType"/>
<element name="MoveObject_InputParams" type="typens:MoveObject_InputParamsType"/>
<element name="PublishObjectNotification_InputParams" type="typens:PublishObjectNotification_InputParamsType"/>
<element name="UpdateAttributes_InputParams" type="typens:UpdateAttributes_InputParamsType"/>
<element name="AddAttributes_InputParams" type="typens:AddAttributes_InputParamsType"/>
<complexType name="SimpleString_ParamsType">
  <sequence>
    <element name="SimpleString" type="string" minOccurs="0">
      <documentation xml:lang="en">
        string element
      </documentation>
    </element>
    <element name="FaultMessage" type="string" minOccurs="0">
      <documentation xml:lang="en">
        FaultMessage
      </documentation>
    </element>
  </sequence>
</complexType>
<complexType name="SimpleBoolean_ParamsType">
  <sequence>
    <element name="SimpleBoolean" type="boolean" minOccurs="0">
      <annotation>
        <documentation xml:lang="en">boolean element</documentation>
      </annotation>
    </element>
    <element name="FaultMessage" type="string" minOccurs="0">
      <annotation>
        <documentation xml:lang="en">FaultMessage</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>

<complexType name="CreateProject_InputParamsType">
  <sequence>
    <element name="uID" type="string">
      <annotation>
        <documentation xml:lang="en">user id of this request</documentation>
        <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
        </appinfo>
      </annotation>
    </element>
    <element name="leadResource" type="string">
      <annotation>
        <documentation xml:lang="en">lead metadata for the new project</documentation>
        <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
        </appinfo>
      </annotation>
    </element>
    <element name="assignNewResourceID" type="boolean">
      <annotation>
        <documentation xml:lang="en">if it is true, mylead agent will assign new resource ID to this project</documentation>
        <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
        </appinfo>
      </annotation>
    </element>
  </sequence>
</complexType>

<complexType name="CreateExperiment_InputParamsType">
  <sequence>
    <element name="uID" type="string">
      <annotation>
        <documentation xml:lang="en">user id of this request</documentation>
        <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
        </appinfo>
      </annotation>
    </element>
  </sequence>
</complexType>
<complexType name="CreateCollection_InputParamsType">
  <sequence>
    <element name="uID" type="string">
      <annotation>
        <documentation xml:lang="en">user id of this request</documentation>
        <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
        </appinfo>
      </annotation>
    </element>
    <element name="leadResource" type="string">
      <annotation>
        <documentation xml:lang="en">lead metadata for the new project</documentation>
        <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
        </appinfo>
      </annotation>
    </element>
    <element name="parentResourceID" type="string">
      <annotation>
        <documentation xml:lang="en">resourceID of the parent node</documentation>
        <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
        </appinfo>
      </annotation>
    </element>
    <element name="assignNewResourceID" type="boolean">
      <annotation>
        <documentation xml:lang="en">if it is true, mylead agent will assign new resource ID to this project</documentation>
        <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
        </appinfo>
      </annotation>
    </element>
  </sequence>
</complexType>
<element name="assignNewResourceID" type="boolean">
    <annotation>
        <documentation xml:lang="en">
            if it is true, mylead agent will assign new resource ID to this project
        </documentation>
        <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
        </appinfo>
    </annotation>
</element>

<complexType name="RegisterFile_InputParamsType">
    <sequence>
        <element name="uID" type="string">
            <annotation>
                <documentation xml:lang="en">
                    user id of this request
                </documentation>
                <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
                </appinfo>
            </annotation>
        </element>
        <element name="leadResource" type="string">
            <annotation>
                <documentation xml:lang="en">
                    lead metadata for the new project
                </documentation>
                <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
                </appinfo>
            </annotation>
        </element>
        <element name="datasourcelocation" type="string">
            <annotation>
                <documentation xml:lang="en">
                    the location of the data source
                </documentation>
                <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
                </appinfo>
            </annotation>
        </element>
        <element name="parentResourceID" type="string">
            <annotation>
                <documentation xml:lang="en">
                    resourceID of the parent node
                </documentation>
                <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
                </appinfo>
            </annotation>
        </element>
        <element name="assignNewResourceID" type="boolean">
            <annotation>
                <documentation xml:lang="en">
                    if it is true, mylead agent will assign new resource ID to this project
                </documentation>
                <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
                </appinfo>
            </annotation>
        </element>
    </sequence>
</complexType>
<complexType name="BulkRegisterFile_InputParamsType">
<sequence>
  <element name="uID" type="string">
    <annotation>
      <documentation xml:lang="en">user id of this request</documentation>
      <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
      </appinfo>
    </annotation>
  </element>
  <element name="leadResource" type="string" maxOccurs="unbounded">
    <annotation>
      <documentation xml:lang="en">lead metadata for the new project as a String array</documentation>
      <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
      </appinfo>
    </annotation>
  </element>
  <element name="datasourcelocation" type="string" maxOccurs="unbounded">
    <annotation>
      <documentation xml:lang="en">the location of the data source as a String array</documentation>
      <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
      </appinfo>
    </annotation>
  </element>
  <element name="parentResourceID" type="string">
    <annotation>
      <documentation xml:lang="en">resourceID of the parent node</documentation>
      <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
      </appinfo>
    </annotation>
  </element>
  <element name="assignNewResourceID" type="boolean" maxOccurs="unbounded">
    <annotation>
      <documentation xml:lang="en">if it is true, mylead agent will assign new resource ID to this project as a boolean array</documentation>
      <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
      </appinfo>
    </annotation>
  </element>
</sequence>
</complexType>

<complexType name="CreateObject_OutputParamsType">
<sequence>
  <element name="ResourceID" type="string" minOccurs="0">
    <annotation>
      <documentation xml:lang="en">resourceID of the new project</documentation>
      <appinfo source="http://www.extreme.indiana.edu/xgws/xwf">
      </appinfo>
    </annotation>
  </element>
</sequence>
</complexType>
<complexType name="BulkCreateObject_OutputParamsType">
<sequence>
  <element name="ResourceID" type="string" minOccurs="0" maxOccurs="unbounded">
    <documentation xml:lang="en">The resourceID of new object as a String array</documentation>
  </element>
  <element name="FaultMessage" type="string" minOccurs="0">
    <documentation xml:lang="en">FaultMessage</documentation>
  </element>
</sequence>
</complexType>
<complexType name="QueryLeadWithContextQuery_InputParamsType">
<sequence>
  <element name="UserDN" type="string">
    <documentation xml:lang="en">The user's DN for this query</documentation>
  </element>
  <element name="Limit" type="int">
    <documentation xml:lang="en">The limit of number of return values</documentation>
  </element>
  <element name="hFilter" type="string">
    <documentation xml:lang="en">The hierarchical filter for this query</documentation>
  </element>
  <element name="cFilter" type="string">
    <documentation xml:lang="en">The context filter for this query</documentation>
  </element>
  <element name="Target" type="string">
    <documentation xml:lang="en">The target for this query</documentation>
  </element>
  <element name="Query" type="string">
    <documentation xml:lang="en">The query for this query</documentation>
  </element>
</sequence>
</complexType>
<complexType name="QueryLead_InputParamsType">
    <sequence>
        <element name="UserDN" type="string">
            <annotation>
                <documentation xml:lang="en">
                    The user's DN for this query
                </documentation>
            </annotation>
        </element>
        <element name="Limit" type="int">
            <annotation>
                <documentation xml:lang="en">
                    The limit of number of return values
                </documentation>
            </annotation>
        </element>
        <element name="hFilter" type="string">
            <annotation>
                <documentation xml:lang="en">
                    The hierarchical filter for this query
                </documentation>
            </annotation>
        </element>
        <element name="cFilter" type="string">
            <annotation>
                <documentation xml:lang="en">
                    The context filter for this query
                </documentation>
            </annotation>
        </element>
        <element name="Target" type="string">
            <annotation>
                <documentation xml:lang="en">
                    The target filter for this query
                </documentation>
            </annotation>
        </element>
    </sequence>
</complexType>

<complexType name="QueryByIds_InputParamsType">
    <sequence>
        <element name="UserDN" type="string">
            <annotation>
                <documentation xml:lang="en">
                    The user's DN for this query
                </documentation>
            </annotation>
        </element>
        <element name="hFilter" type="string">
            <annotation>
                <documentation xml:lang="en">
                    The hierarchical filter for this query
                </documentation>
            </annotation>
        </element>
        <element name="cFilter" type="string">
            <annotation>
                <documentation xml:lang="en">
                    The context filter for this query
                </documentation>
            </annotation>
        </element>
        <element name="Target" type="string">
            <annotation>
                <documentation xml:lang="en">
                    The target filter for this query
                </documentation>
            </annotation>
        </element>
    </sequence>
</complexType>
<complexType name="ResourceId" type="string" maxOccurs="unbounded">
    <annotation>
        <documentation xml:lang="en">
            The resourceIDs for this query
        </documentation>
    </annotation>
</complexType>

<complexType name="QueryById_InputParamsType">
    <sequence>
        <element name="UserDN" type="string">
            <annotation>
                <documentation xml:lang="en">
                    The user's DN for this query
                </documentation>
            </annotation>
        </element>
        <element name="hFilter" type="string">
            <annotation>
                <documentation xml:lang="en">
                    The hierarchical filter for this query
                </documentation>
            </annotation>
        </element>
        <element name="cFilter" type="string">
            <annotation>
                <documentation xml:lang="en">
                    The context filter for this query
                </documentation>
            </annotation>
        </element>
        <element name="ResourceIDs" type="string">
            <annotation>
                <documentation xml:lang="en">
                    The resourceIDs for the query
                </documentation>
            </annotation>
        </element>
    </sequence>
</complexType>

<complexType name="QueryWorkspace_InputParamsType">
    <sequence>
        <element name="UserDN" type="string">
            <annotation>
                <documentation xml:lang="en">
                    User's DN for this query.
                </documentation>
            </annotation>
        </element>
        <element name="resourceId" type="string">
            <annotation>
                <documentation xml:lang="en">
                    The resourceID
                </documentation>
            </annotation>
        </element>
        <element name="limit" type="int">
            <annotation>
                <documentation xml:lang="en">
                    limit of returning values. 0 is the default value.
                </documentation>
            </annotation>
        </element>
    </sequence>
</complexType>

<complexType name="QueryWorkspaceStructure_InputParamsType">
    <sequence>
    </sequence>
</complexType>
<element name="UserDN" type="string">
  <annotation>
    <documentation xml:lang="en">
      user's DN of this query. This query will return whole structure of the user's workspace
    </documentation>
  </annotation>
</element>
</complexType>

<complexType name="Query_OutputParamsType">
  <sequence>
    <element name="Query_Result" type="string">
      <annotation>
        <documentation xml:lang="en">
          result of the query. for creating objects, this is the new resource id. for searching query, this is the result of searching activity. for register file query, this is the destination file URL.
        </documentation>
      </annotation>
    </element>
    <element name="FaultMessage" type="string">
      <annotation>
        <documentation xml:lang="en">
          FaultMessage
        </documentation>
      </annotation>
    </element>
  </sequence>
</complexType>

<complexType name="AddNewUser_InputParamsType">
  <sequence>
    <element name="AdminDN" type="string">
      <annotation>
        <documentation xml:lang="en">
          the userDN for the administrator
        </documentation>
      </annotation>
    </element>
    <element name="NewUserInfo" type="typens:MyLeadUser_ParamsType">
      <annotation>
        <documentation xml:lang="en">
          Information of the new user
        </documentation>
      </annotation>
    </element>
  </sequence>
</complexType>

<complexType name="MyLeadUser_ParamsType">
  <sequence>
    <element name="DN" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation xml:lang="en">
          DN of the new user
        </documentation>
      </annotation>
    </element>
    <element name="Name" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation xml:lang="en">
          Name of the new user
        </documentation>
      </annotation>
    </element>
    <element name="Replica" type="string" minOccurs="0" maxOccurs="1">
      <annotation>
        <documentation xml:lang="en">
          Replica nickname the new user is assigned to
        </documentation>
      </annotation>
    </element>
  </sequence>
</complexType>
<complexType name="MyLeadStorage_ParamsType">  
  <sequence>  
    <element name="GlobalID" type="string" minOccurs="0" maxOccurs="1">  
      <documentation xml:lang="en">Global ID of the storage resource</documentation>  
    </element>  
    <element name="AccessURL" type="string" minOccurs="0" maxOccurs="1">  
      <documentation xml:lang="en">Access URL of the storage resource</documentation>  
    </element>  
    <element name="Nickname" type="string" minOccurs="0" maxOccurs="1">  
      <documentation xml:lang="en">Nickname of the storage resource</documentation>  
    </element>  
    <element name="Protocol" type="string" minOccurs="0" maxOccurs="unbounded">  
      <documentation xml:lang="en">Protocol of the storage resource</documentation>  
    </element>  
  </sequence>  
</complexType>
transport protocols of the storage resource

<complexType name="DeleteObject_InputParamsType">
  <sequence>
    <element name="DN" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation xml:lang="en">
          DN of the owner of the object to be deleted.
        </documentation>
      </annotation>
    </element>
    <element name="GlobalID" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation xml:lang="en">
          Global ID of the object to be deleted
        </documentation>
      </annotation>
    </element>
  </sequence>
</complexType>

<complexType name="DeleteObject_OutputParamsType">
  <sequence>
    <element name="Results" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation xml:lang="en">
          Results of the delete object operation
        </documentation>
      </annotation>
    </element>
    <element name="FaultMessage" type="string">
      <annotation>
        <documentation xml:lang="en">
          FaultMessage
        </documentation>
      </annotation>
    </element>
  </sequence>
</complexType>

<complexType name="MoveObject_InputParamsType">
  <sequence>
    <element name="DN" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation xml:lang="en">
          DN of the owner of the object to be moved
        </documentation>
      </annotation>
    </element>
    <element name="GlobalID" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation xml:lang="en">
          Global ID of the object to be moved
        </documentation>
      </annotation>
    </element>
    <element name="NewParentID" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation xml:lang="en">
          Resource ID of the new parent
        </documentation>
      </annotation>
    </element>
  </sequence>
</complexType>

<complexType name="MoveObject_OutputParamsType">
<sequence>
  <element name="Results" type="string">
    <annotation>
      <documentation xml:lang="en">Results of the move operation</documentation>
    </annotation>
  </element>
  <element name="FaultMessage" type="string">
    <annotation>
      <documentation xml:lang="en">FaultMessage</documentation>
    </annotation>
  </element>
</sequence>

<complexType name="PublishNotificationTypeEnumeration">
  <restriction base="xsd:string">
    <enumeration value="publishSucceeded" />
    <enumeration value="publishFailed" />
  </restriction>
</complexType>

<complexType name="PublishObjectNotification_InputParamsType">
  <sequence>
    <element name="DN" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation xml:lang="en">DN of the user</documentation>
      </annotation>
    </element>
    <element name="notificationType" type="typens:PublishNotificationTypeEnumeration" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation xml:lang="en">Type of the publish notification message</documentation>
      </annotation>
    </element>
    <element name="resourceID" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation xml:lang="en">LEAD Resource ID</documentation>
      </annotation>
    </element>
    <element name="message" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation xml:lang="en">The message of the notification</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>

<complexType name="PublishObjectNotification_OutputParamsType">
  <sequence>
    <element name="Results" type="string">
      <annotation>
        <documentation xml:lang="en">Results of sending the publish notification message</documentation>
      </annotation>
    </element>
    <element name="FaultMessage" type="string">
      <annotation>
        <documentation xml:lang="en"></documentation>
      </annotation>
    </element>
  </sequence>
</complexType>
Fault Message
</documentation>
</element>
</sequence>
</complexType>

<complexType name="UpdateAttributeHolderType">
    <sequence>
        <element name="attributeXml" type="string" minOccurs="1" maxOccurs="1">
            <annotation>
                <documentation xml:lang="en">Revised attribute in IMS xml fragment</documentation>
            </annotation>
        </element>
        <element name="guid" type="string" minOccurs="1" maxOccurs="1">
            <annotation>
                <documentation xml:lang="en">GUID of the object to which this attribute belong</documentation>
            </annotation>
        </element>
        <element name="position" type="int" minOccurs="0" maxOccurs="1">
            <annotation>
                <documentation xml:lang="en">Position of the element if multiple instances are present. Omit if only single instance is allowed.</documentation>
            </annotation>
        </element>
        <element name="timestamp" type="long" minOccurs="0" maxOccurs="1">
            <annotation>
                <documentation xml:lang="en">Timestamp accompanying the attribute of the format YYYYMMDDhhmmss. If present, myLEAD checks to make sure the updated timestamp is later than the stored one. If omitted, update occurs no matter what.</documentation>
            </annotation>
        </element>
    </sequence>
</complexType>

<complexType name="UpdateAttributes_InputParamsType">
    <sequence>
        <element name="DN" type="string" minOccurs="1" maxOccurs="1">
            <annotation>
                <documentation xml:lang="en">DN of the user</documentation>
            </annotation>
        </element>
        <element name="Updates" type="typens:UpdateAttributeHolderType" minOccurs="1" maxOccurs="unbounded">
            <annotation>
                <documentation xml:lang="en">An array of attributes to be updated</documentation>
            </annotation>
        </element>
    </sequence>
</complexType>

<complexType name="UpdateAttributes_OutputParamsType">
    <sequence>
        <element name="Results" type="string">
            <annotation>
                <documentation xml:lang="en">Results of updating attributes</documentation>
            </annotation>
        </element>
        <element name="FaultMessage" type="string">
            <annotation>
                <documentation xml:lang="en"></documentation>
            </annotation>
        </element>
    </sequence>
</complexType>
<complexType name="AddAttributes_InputParamsType">
  <sequence>
    <element name="DN" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation xml:lang="en">DN of the user</documentation>
      </annotation>
    </element>
    <element name="guid" type="string" minOccurs="1" maxOccurs="1">
      <annotation>
        <documentation xml:lang="en">GUID of the object to which the attributes belong</documentation>
      </annotation>
    </element>
    <element name="attributesXml" type="string" minOccurs="1" maxOccurs="unbounded">
      <annotation>
        <documentation xml:lang="en">An array of attributes to be added in LMS XML format</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>

<complexType name="AddAttributes_OutputParamsType">
  <sequence>
    <element name="Results" type="string">
      <annotation>
        <documentation xml:lang="en">Results of adding attributes</documentation>
      </annotation>
    </element>
    <element name="FaultMessage" type="string">
      <annotation>
        <documentation xml:lang="en">Fault Message</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>