The Manual for CNRI Digital Object Registry

Version 3.5

Prepared by:
Corporation for National Research Initiatives
Acknowledgment

The previous version of this manual was a collaborative effort between Booz Allen Hamilton and Corporation for National Research Initiatives (CNRI). CNRI gratefully acknowledges the work of Booz Allen Hamilton.

This work has been funded in part by the Advanced Distributed Learning (ADL) Initiative and the Defense Advanced Research Projects Agency (DARPA), through Grant No. HR0011-05-0003.
# Table of Contents

**THE MANUAL FOR CNRI DIGITAL OBJECT REGISTRY**

**VERSION 3.5**

## 1 INTRODUCTION

1.1 Purpose & Scope

1.2 Intended Audience

1.3 Registry Overview

1.4 Support

## 2 REFERENCED DOCUMENTS AND WEB SITES

## 3 INSTALLATION CHECKLIST

## 4 REQUIREMENTS

4.1 Hardware

4.1.1 Architecture

4.1.2 Main Memory

4.1.3 Hard Disk Space

4.2 Software

4.2.1 Operating System

4.2.2 Java SE Development Kit (JDK) 5.0

4.2.3 Apache Ant

4.2.4 Application Servers

4.2.5 SMTP Server

4.2.6 Handle System®

4.2.7 CNRI DO Repository

4.2.8 Database with JDBC Support

4.2.9 Virus Scanner

4.2.10 Third party Libraries

## 5 INSTALLATION INSTRUCTIONS

5.1 Operating System

5.2 Unprivileged User Account

5.3 Install Required Software

5.4 Database Installation

5.5 Handle Server Installation

5.6 Handle Server Prefix Configuration

5.7 CNRI Repository Installation
5.8 Download the Registry Software ............................................................................................................. 22
5.9 Creating users and Groups .......................................................................................................................... 23
5.10 Using Custom Validation Module .................................................................................................................. 24
5.11 Using Custom Transformation Module ...................................................................................................... 26
5.12 Configuring the Registry ............................................................................................................................... 27
5.13 Update HTML files with your Registry Handle ........................................................................................... 38
5.14 Installing CNRI Repository Software ......................................................................................................... 40
   5.14.1 Deploying Core Registry ......................................................................................................................... 40
   5.14.2 Deploying Index Engine ........................................................................................................................... 40
   5.14.3 Deploying RIM-LITE ............................................................................................................................... 40

6 Registry Startup and Shutdown ....................................................................................................................... 41
   6.1.1 Handle Server Startup and Shutdown ....................................................................................................... 41
   6.1.2 Database Startup and Shutdown ............................................................................................................... 41
   6.1.3 DO Repository Startup & Shutdown ......................................................................................................... 41
   6.1.4 Web Container Startup and Shutdown .................................................................................................... 42

7 Post Installation Validation ............................................................................................................................... 42
   7.1 Testing the Core Registry ............................................................................................................................. 42
   7.2 Testing the Registry using RIM-LITE ............................................................................................................ 44

8 Appendix A – Overview of the Handle Admin Tool ......................................................................................... 48
   8.1 Starting the Handle Admin Tool ..................................................................................................................... 48
   8.2 Authenticating to the Handle System ............................................................................................................. 49
   8.3 Creating or Modifying a Handle ..................................................................................................................... 52
1 Introduction

Corporation for National Research Initiatives (CNRI) has developed the Digital Object Registry with support from the Advanced Distributed Learning (ADL) Initiative. It provides the infrastructure necessary for registering and discovering content objects in the learning, education and training domain. For the sake of brevity, we will refer to the ‘CNRI Digital Object Registry’ as the ‘Registry’ in the following sections of this document.

The Advanced Distributed Learning (ADL) Initiative funded CNRI to develop CORDRA™, which is an open standards based model for designing and developing software systems for the purpose of discovery, sharing and reuse of content objects.

The ADL Registry (ADL-R) is the first publicly available Content Object Repository Discovery and Registration (CORDRA) registry. ADL-R provides a registry of content objects for the DoD and encourages their discovery and, given appropriate permissions, re-use by the DoD community as well as the general public.

The ADL Registry was designed primarily for the DoD community, specifically that part of the community interested in learning objects. CNRI is generalizing the basic technology, with continued support from ADL and others, to create a general-purpose registry that can be used to register any type of material for any community of practice.

The current DO Registry (version 3.5) is configurable in a number of ways, but is still fairly ADL-R centric and is recommended only to those who wish to clone the ADL-R for their community. As the generalization process continues and future versions are released, the DO Registry will become increasingly less oriented to the world of learning objects and the ADL-R procedures and will be configurable to enable the registration and management of any type of objects for any purpose. This process should be complete before the end of 2008. See the Registry website at http://doregistry.org/ for details and updates.

1.1 Purpose & Scope

This document provides a step-by-step approach to follow to install a version of DO Registry 3.5 for a community of practice.

This document does not go into the details of installation of required hardware and software. These requirements are documented below and there are several options to meet each of these requirements. Please refer to product specific documentation of the option chosen, to meet the requirement.
### 1.2 Intended Audience

This document is targeted to a technical audience. The reader is expected to have an understanding of:

- Java Enterprise technologies and Application Servers
- System Administration of the chosen Operating System
- Networks and protocols
- Databases and Database Administration
- XML and related technologies

It is highly recommended that readers familiarize themselves with the concepts of registration and discovery of content objects in the learning domain. The “Referenced Documents and Web Sites” section lists the documents that address these concepts.

### 1.3 Registry Overview

The Registry software is located at the CNRI Digital Object Registry Distribution website. [http://doregistry.org/download.html](http://doregistry.org/download.html)

The Registry software consists of following major components:

- Core Registry Software
- RIM-LITE component that provides an external interface to the Core Registry

In addition, the Registry depends on the Handle System and the CNRI Digital Object (DO) Repository software, which are freely available but must be downloaded separately.

The key components of the Registry, their inter-relationships and dependencies are depicted in the high-level architecture diagram below:
1.4 Support

The installer can use the following contact information for any clarifications or issues encountered during the Registry installation process:

- Support for Registry and DO Repository is available at: http://doregistry.org/support.html
- Support for Handle System is available at: hdladmin@cnri.reston.va.us

2 Referenced Documents and Web Sites

1. http://adlregistry.adlnet.gov/. This is the ADL Registry Portal and provides access to various ADL-R functions and documentation.
2. http://www.handle.net. This is the Handle System website that provides access to the Handle System documentation and the software distribution.
3. "Handle System Overview", http://hdl.handle.net/4263537/4069

3 Installation Checklist

This high level checklist below may be used to plan and execute the Registry Installation Process.
Table 1. Installation Checklist

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>Review the information from the section on “Referenced Documents and Web Sites” to develop an understanding of the Registry and the Handle System.</td>
</tr>
<tr>
<td>Step 2</td>
<td>Review the Hardware and Software Requirements from the “Requirements” Section below.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Setup a machine with an appropriate Operating System.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Create an unprivileged account which will be used to setup and run the Registry.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Install Java SE Development Kit (JDK) 5.0.</td>
</tr>
<tr>
<td>Step 6</td>
<td>Install Apache Ant.</td>
</tr>
<tr>
<td>Step 7</td>
<td>Install a J2EE 1.4 Servlet and JSP compliant Container (e.g. Tomcat 5).</td>
</tr>
<tr>
<td>Step 8</td>
<td>Install a command line virus scanner (optional).</td>
</tr>
<tr>
<td>Step 9</td>
<td>Install the Handle Server.</td>
</tr>
<tr>
<td>Step 10</td>
<td>Register with CNRI Handle System Administrator and obtain a Naming Prefix.</td>
</tr>
<tr>
<td>Step 11</td>
<td>Install CNRI Repository.</td>
</tr>
<tr>
<td>Step 12</td>
<td>Obtain Registry Distribution from CNRI.</td>
</tr>
<tr>
<td>Step 13</td>
<td>Configure user accounts and Registry.</td>
</tr>
<tr>
<td>Step 14</td>
<td>Install Registry, Indexing Engine and RIM-LITE.</td>
</tr>
<tr>
<td>Step 15</td>
<td>Test the Core Registry.</td>
</tr>
<tr>
<td>Step 16</td>
<td>Test Core Registry and RIM-LITE combined, via the RIM-LITE interface to the Registry.</td>
</tr>
</tbody>
</table>

4 Requirements

4.1 Hardware

4.1.1 Architecture

The Registry is tested to run on:

- Sun Fire V440 with 64-bit SPARC v9 architecture
- Intel Xeon 64-bit architecture
• AMD Opteron Systems

Although not tested on other platforms, the Registry is designed as platform-independent software.

4.1.2 Main Memory

Minimum Required: 2GB
Minimum Recommended: 4GB

4.1.3 Hard Disk Space

Minimum Required for Installation: 100MB
Minimum Recommended: 20 GB

It was observed that in a Test ADL Registry, 3000 items required 10MB of storage. These numbers can be used to estimate the order of magnitude of storage space required.

The specific storage space needed for your instance of the Registry may vary considerably. The actual requirement for disk space depends on the amount of data that will be eventually stored in the Registry.

4.2 Software

4.2.1 Operating System

The Registry is tested to run on:
• SunOS 5.9
• Fedora Core 4 SMP 2.6.13
• Red Hat Enterprise Linux AS Release 4.5
• Ubuntu Linux 7.04

4.2.2 Java SE Development Kit (JDK) 5.0

JDK 5.0 is required to compile and run the software. The software may be downloaded from http://java.sun.com/.
Note: When working with a 64-bit or 32-bit Operating System, all components, especially the JDK, should also be for that platform. The Registry installation will behave erratically if that is not the case.

4.2.3 Apache Ant

Apache ANT is required to compile the software. The software may be downloaded from [http://ant.apache.org/](http://ant.apache.org/). As of this writing, Ant 1.7.0 is the latest released version. Ant 1.7.0 was used to build the Registry software.

4.2.4 Application Servers

The Registry is tested to run on the following J2EE 1.4 compliant Servlet & JSP Containers:

- Apache Tomcat 5.5.25
- JBoss Application Server
- BEA Weblogic Express 9.1

4.2.5 SMTP Server

An SMTP server, installed on any machine, needs to be made accessible to the Registry. If an existing SMTP server is available, a new SMTP server need not be installed.

The SMTP server is needed by the registry to send out emails when certain registry transactions are performed.

4.2.6 Handle System®

A Handle Server is required for homing the prefixes related to the Registry. We recommend installing the Handle Server on the same machine (or the same network) as the Registry for the best performance.

We also recommend that the latest version of the Handle Server be installed. It will generally be available from [http://www.handle.net](http://www.handle.net). At the time of this writing, the latest release is HANDLE.NET® version 6.2.5. The Registry installation needs version 6.2.5 or higher, as this version supports the password obscurity feature.
4.2.7 CNRI DO Repository

The Registry has been designed to store registration records in the CNRI DO Repository. We recommend installing the Repository on the same machine (or in the same network) as the Registry for the best performance.

Version 1.6 of the Repository must be used. It may be obtained at http://dorepository.org/index.html. An example installation specific to its use in conjunction with the Registry is given below.

4.2.8 Database with JDBC Support

RIM-LITE provides a SQL interface to an underlying database. The SQL interface support is an optional feature in RIM-LITE. The database is used for storing user registrations and repository registration information. Java Database Connectivity (JDBC) support to the database is expected by the corresponding modules, if used, inside RIM-LITE.

For the purposes of this Installation Document, MySql 4.1.22 was used. However, any other Relational Database Management System (RDBMS) that supports JDBC may be used instead.

4.2.9 Virus Scanner

Submissions to the registry may be screened for viruses. A command line virus scanner, if present, will be used by RIM-LITE.

This Installation Document shows the example steps using “McAfee VirusScan Command Line Scanner for Linux 5.20”.

4.2.10 Third party Libraries

The following table describes the third party libraries used by the Registry software, including URLs to the appropriate websites. However, we highly recommend using the libraries made available through the CNRI Digital Object Registry Distribution website, as the software is tested to operate correctly with these specific libraries. The libraries that are available from corresponding websites may have changed over time.
### Table 2. Third Party Developed Libraries

<table>
<thead>
<tr>
<th>Third Party Developed Libraries</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apache Lucene 1.4.3</td>
<td><a href="http://lucene.apache.org/java/docs/index.html">http://lucene.apache.org/java/docs/index.html</a></td>
</tr>
<tr>
<td>Jaxen Core</td>
<td><a href="http://jaxen.codehaus.org/releases.html">http://jaxen.codehaus.org/releases.html</a></td>
</tr>
<tr>
<td>Jaxen JDOM</td>
<td><a href="http://jaxen.codehaus.org/releases.html">http://jaxen.codehaus.org/releases.html</a></td>
</tr>
<tr>
<td>JDOM 1.0</td>
<td><a href="http://www.jdom.org/">http://www.jdom.org/</a></td>
</tr>
<tr>
<td>JFreeChart 1.0.1</td>
<td><a href="http://www.jfree.org/jfreechart/">http://www.jfree.org/jfreechart/</a></td>
</tr>
<tr>
<td>JCommon 1.0.1</td>
<td><a href="http://www.jfree.org/jcommon/index.php">http://www.jfree.org/jcommon/index.php</a></td>
</tr>
<tr>
<td>SAXPath 1.0</td>
<td><a href="http://sourceforge.net/projects/saxpath/">http://sourceforge.net/projects/saxpath/</a></td>
</tr>
<tr>
<td>Apache Commons Codec</td>
<td><a href="http://commons.apache.org/codec/">http://commons.apache.org/codec/</a></td>
</tr>
<tr>
<td>Apache Commons HttpClient</td>
<td><a href="http://hc.apache.org/httpclient-3.x/">http://hc.apache.org/httpclient-3.x/</a></td>
</tr>
<tr>
<td>Apache Commons Logging</td>
<td><a href="http://commons.apache.org/logging/">http://commons.apache.org/logging/</a></td>
</tr>
<tr>
<td>Saxon</td>
<td><a href="http://saxon.sourceforge.net/">http://saxon.sourceforge.net/</a></td>
</tr>
</tbody>
</table>

### 5 Installation Instructions

#### 5.1 Operating System

For the purpose of documenting the Installation Steps, we use Red Hat Enterprise Linux AS Release 4.5 as an example. However, the Registry has been tested to run on several Operating Systems, as listed in the “Requirements” section. Your choice of Operating System may depend on factors such as the existing Operating Systems in use in your community of practice, the cost, existing technical expertise, etc.

**Note:** Please pay special attention to whether your Operating System is 32-bit or 64-bit. All components installed on a given Operating System should respectively be 32-bit or 64-bit. Specifically, the JDK should correspond to the same bit size. If that is not the case, the Registry Installation will behave erratically.

#### 5.2 Unprivileged User Account

The Registry can be installed and operated using an unprivileged user account. On a multi-user system such as Unix based systems, the Registry can be run using the superuser (root) account or as an unprivileged user account. Security best practices recommend that the principle of ‘least privileges’ be applied in such situations, so that processes run under the minimum privileges needed to operate.
As an example, we have created an unprivileged user account as shown below:

User id: adl
Group id: adl
Home Directory: /home/adl/
Minimal Permissions on /home/adl:
   adl@/home/adl>ls -ld .
   drwx------ 25 adl adl 4096 Oct 15 13:58 ./

From this point onward, <adl_home> represents the home directory for the user account created for the purpose of installing and running the Registry.

5.3 Install Required Software

1. Install Java SE (JDK) 5.0. Please follow the instructions specific to your platform. As an example, we installed JDK 5.0 to the location: /usr/local/jdk1.5.0_12/

2. Setup the environment variable JAVA_HOME to point to the JDK 5.0 installation, and validate the same. For example, we validated using the command shown below:

   adl@/home/adl>echo $JAVA_HOME
   /usr/local/jdk1.5.0_12

3. Add $JAVA_HOME/bin to your PATH environment variable.

4. Install Apache Ant. For example, we installed Ant to the location /usr/local/apache-ant-1.7.0/

5. Setup the environment variable ANT_HOME such that it points to the installation location of Apache Ant. For example:

   adl@/home/adl>echo $ANT_HOME
   /usr/local/apache-ant-1.7.0

6. Add $ANT_HOME/bin to your PATH environment variable.
7. Install the Application Server. Please follow the instructions specific to your choice of the Application Server, and validate that the installation was successful. For example, we installed Apache Tomcat under /usr/local/registry/apache-tomcat-5.5.25/.

8. Install a command line Virus Scanner if it is used. The virus scanner is assumed to be at the location /usr/local/bin/uvscan. If this needs to be changed, please contact CNRI. As an example, we installed the “McAfee VirusScan Command Line Scanner for Linux 5.20” at the location shown below:

```
[adl@atlas-vmsrv02 tmp/vlp4520]
$ ./install-uvscan
Which directory do you want to install into? [/usr/local/uvscan]
/usr/local/uvscan doesn't exist. Create it? [y]/n  y
Do you want to create the link(s) to uvscan in /usr/local/bin [y]/r
Do you want to create the link(s) to uvscan_secure in /usr/local/bin [y]/n
Do you want to create the link(s) to liblnx5v.so.4 in /usr/local/lib [y]/n
Do you want to create the link(s) to uvscan,1 in /usr/local/man/man1 [y]/n
/usr/local/man/man1 doesn't exist, Create it? [y]/n

installation complete.

Do you want to perform a scan of all filesystems y/[n] n
```

Figure 2. Command Line Virus Scanner Installation

```
adl@/usr/local>ls -1d uvscan
drwxr-xr-x  2 root root 4096 Sep 21 16:20 uvscan/

adl@/usr/local/bin>ls -l
total 8
lrwxrwxrwx  1 root root 24 Oct 18 15:37 uvscan -> /usr/local/uvscan/uvscan*
```
Please ensure that you are using latest virus definition files provided by your vendor. Most Virus Scanners require the virus data definition files to be periodically updated for continued protection against new viruses.

### 5.4 Database Installation

RIM-LITE provides access to a database to support any Registry related operations such as User Registration and Repository Registration.

You may use any JDBC compliant database for your installation. The corresponding JDBC java driver must be placed in the RIM/WebRoot/WEB-INF/lib directory in the downloaded registry software. Please refer to the Database specific documentation for detailed steps for installing and administering the database.

We installed MySql 4.1.22 as an example. Below are some of the specific configurations we setup for MySql:

- MySql was installed under `/usr/local/` directory.
- As a security best practice, we did not install and run MySql with root privileges. Instead, we created a user account of `mysqladm` with group membership to `mysqlgrp` to run MySql.
- We setup a database `rimlitedb` for RIM-LITE.
- We followed MySql administration steps to setup user accounts. Specifically, we setup a user `rimlite` with access to the `rimlitedb`.
- MySql was setup as a service so MySql would automatically startup and gracefully shutdown with the machine startup and shutdown.
- We obtained the MySql JDBC driver for use by RIM-LITE.

### 5.5 Handle Server Installation

Instructions to download and install the Handle Server software are available at [http://www.handle.net/](http://www.handle.net/). We recommend, for performance reasons, that the Handle Server be installed on the same machine (or on the same network) as the Registry.

For our example, we installed the Handle Server as below:

- Handle server was installed on the same machine on which Registry was installed.
- We used the same user id ‘adl’, as was used to install the Registry, to install and run the Handle Server.
- The Handle Server was installed at location: `<adl_home>/hdl6.2.5_01/`
- The Handle Server was made accessible over the Internet. Please ensure that the ports you have chosen for the Handle Server are accessible over the Internet and no firewalls are preventing access to the Handle Server.
- If your machine has an internal IP (behind firewall), which is different from the external IP, please ensure that you provide the external IP, as shown below, during installation.

![Figure 3. Handle Server Installation](image)

### 5.6 Handle Server Prefix Configuration

To complete the Handle Server Configuration, a ‘prefix’ needs to be requested from the CNRI Handle System Administrator. Complete details about requesting a prefix are given at [http://www.handle.net/download.html](http://www.handle.net/download.html).
If you have already been allocated a handle prefix, for example, if this were a re-installation of the software, or if you are already running a handle server for some other project, this step may be omitted. In such a situation, please use this pre-existing handle prefix to complete the configuration of the Handle Server, as explained in the documentation that accompanies the Handle System software.

In this document we will use the prefix of 10233 as an example. This is the prefix that was assigned by the CNRI Handle System Administrator. For your installation, please replace 10233 with the value assigned to you by the CNRI Handle System Administrator.

5.7 CNRI Repository Installation

The instructions for downloading and installing the CNRI DO Repository software are available at http://www.cnri.reston.va.us/repository/. We recommend installing the Repository on the same machine (or the same network) as the other Registry components, for performance reasons.

For our example, we installed the CNRI Repository by following the instructions below:

1. Extract the do-1.2.tar.gz. For example,
   adl@/usr/local/registry>tar xvf /home/adl/CDOR/DO/dist/do-1.2.tar

2. Put the bin directory from the package in your PATH environment variable. For example, we added /usr/local/registry/do-1.2/bin to our PATH.

3. Create the data directory for data storage. For example,
   adl@/usr/local/registry>mkdir data

4. Run the command: do-server -setup <serverdir> where <serverdir> is a directory for the server configuration and data storage. For our example, <serverdir> is /usr/local/registry/data as shown below:
Note: The DO Repository Server needs to be accessible only to the Registry. For security reasons we recommend that firewall settings should be in place to prevent the DO Repository Server from being accessible over the Internet.

When asked for a repository identifier, provide the handle used to identify the repository i.e: <the-prefix>/repository. For example, we provided 10233/repository.

When asked for an identifier to distinguish this server from others in the same service enter the sub-prefix you intend to use: i.e: pr
The output of this command will be similar to the example shown below:

![Example output](image)

Figure 5. CNRI Repository Installation Completion

5. After the set up is finished, create the repository handle `<the-prefix>/repository`.

The handle record has the following data

- **CNRI.OBJECT_SERVER_INFO** value: Contents of the `server_info.xml` file that is located in `<server-dir>/server_info.xml`. 
- **CNRI.OBJECT_SERVER** value: `<the-prefix>/repository`  
- **HS_PUBKEY** by loading the public key file from `<serverdir>/publickey` into it

For example, we created the repository handle 10233/repository as shown:

![Repository Handle Creation](image)

**Figure 6. Repository Handle Creation**

6. Set the default permissions for objects in the repository. Create the file called `<server_dir>/default_rights.txt` with the following lines:

```plaintext
accept:* <the-prefix>/repository
accept:* <the-prefix>/registry
accept:* [0.na/<the-prefix>] *
accept:* [0.NA/<the-prefix>] *
accept:1037/0 * *
accept:1037/4 * *
accept:1037/5 * *
accept:1037/8 * *
accept:1037/9 * *
```
**Note:** Please do not copy and paste from this document to your `default_rights.txt` file. Type in the above lines to the `default_rights.txt` file, as the characters between fields above are single TAB characters and not spaces. The permissions shown in this file indicate that your repository can perform any operation on itself, and that the registry can perform any operation on the repository. All ‘safe’ operations, e.g., listing data elements, viewing data elements, or inspecting objects, can be performed by any client.

For our example, the location and contents of the `default_rights` is shown below:

![Example default_rights.txt file](image)

**Figure 7. Example default_rights.txt file**

7. Then run the command: `do-server <serverdir>`. This starts the server, and if this server is part of a multi-server repository, will allow the replication process to begin.

   For example, we start the server with the command:
   
   adl@/usr/local/registry>do-server /usr/local/registry/data/ &

8. If you are remotely installing on a Unix/Linux machine, you’ll need to have an X Server running on your local workstation for the next step to complete. The remote Unix/Linux machine will need the ability to connect to the X Server running on your local workstation and display a window.

9. Now tell the repository that these are the default rights for any objects in the repository by running the following command.

   do-involve [repository handle] 1037/6 elementid=internal.default_rights < default_rights.txt
A window will appear asking you to authenticate. In the authentication window, enter the identifier of your repository (\textless\texttt{the-prefix}>/repository) and select the private key file in the directory of your repository. The command should then complete, setting the rights for the repository object.

The example below shows the expected interaction:

```
adl@/usr/local/registry>do-involve 10233/repository 1037/6 elementid=internal.default_rights <data/default_rights.txt
Enter Operation Input, then press Control-D (Control-Z on Windows)
response:status=success
```

![Authentication Details](image)

**Figure 8. Configuring the Default Rights for the Repository Objects**

10. Use the \texttt{do-client} application to talk to your new digital object server and configure access control and authorization. Without further configuration, the access/authentication system will only allow access to the client that authenticated using the same handle ID as the server.

5.8 **Download the Registry Software**


2. Unzip CDOR.zip to \textless\texttt{adl_home}\texttt{>. This should result in the directory structure under \textless\texttt{adl_home}\texttt{/CDOR as shown below:}
5.9 Creating users and Groups

A simple tool is bundled into this release that creates users and groups. Figure 10 shows an example of a user and group creation. Note that if the group already exists, the tool adds the user to the group.

For the ADL-R Practice and Operational Registry, an ADL specific User Registration Tool (URT) is used to create users and groups. This tool acts as a client to both the Handle System and a database (through RIM) that creates the users and groups in these systems. In this approach, the database stores additional information about users and groups that are not stored in the Handle System. For more detail about URT, please contact CNRI.
Figure 10. User Registration Tool Creating a User and a Group

5.10 Using Custom Validation Module

The built-in validation module may be replaced with a custom validation module. Any submission to the current version of the Registry, however, must be valid LOM (Learning Object Metadata). Any custom validation must accommodate any LOM extensions or modifications. The following code listing provides a skeleton code for writing a custom validation module.
1: package sample.registry;
2: 
3: import net.cnri.adlr.rim.serviceimpl.preprocess.Chain;
4: import net.cnri.registry.rim.RIMContext;
5: import net.cnri.registry.rim.RIMException;
6: import net.cnri.registry.rim.RIMMessage;
7: import net.cnri.registry.rim.serviceimpl.RIMPreprocessor;
8: 
9: public class NewValidation extends Chain {
10: 
11: public NewValidation(RIMPreprocessor preprocessor) {
12: super(preprocessor);
13: }
14: 
15: public RIMMessage preprocess(RIMMessage message, RIMContext context)
16: throws RIMException {
17: 
18: // Validation code here.
19: // ByteArrayInputStream bais =
20: // message.typedObjectStream.getByteArrayInputStream();
21: // Document doc = parse(bais);
22: // validate(doc);
23: 
24: // Throw a RIMException if there is a validation error
25: 
26: return super.callTrailer(message, context);
27: }
28: 
29: }

Listing 1: New validation module
Compile the code above using the java libraries from the RIM/WebRoot/WEB-INF/lib/ directory and place the compiled library into the same directory.

5.11 Using Custom Transformation Module

The built-in transformation module may be replaced with a custom transformation module, either to support different audience profiles (PDA etc), or to enhance the existing one. The following code listing provides a skeleton code for writing a custom transformation module.

```java
package sample.registry;

import net.cnri.registry.rim.RIMContext;
import net.cnri.registry.rim.RIMException;
import net.cnri.registry.rim.RIMMessage;
import net.cnri.registry.rim.serviceimpl.RIMPostprocessor;

public class NewTransformation extends Chain {

    public NewTransformation(RIMPostprocessor postprocessor) {
        super(postprocessor);
    }

    public RIMMessage postprocess(RIMMessage message, RIMContext context) throws RIMException {
        // Transformation code here
        // ByteArrayInputStream bais =
        // message.typedObjectStream.getByteArrayInputStream();
        // Based on the different values of message.originalType, the
        // transformation method may be applied.
        // Document transformedResponse = transform(response,
        // message.originalType);
        // ByteArrayOutputStream bytesout = toBytes(transformedResponse);

        // ByteArrayInputStream bais = message.typedObjectStream.getByteArrayInputStream();
        // Based on the different values of message.originalType, the
        // transformation method may be applied.
        // Document transformedResponse = transform(response,
        // message.originalType);
        // ByteArrayOutputStream bytesout = toBytes(transformedResponse);
    }
```
26: // TypedObjectStream newTOS = new TypedObjectStream(null,
27: // mimeType, “Output”, bytesout);
28: // message.typedObjectStream = newTOS;
29: return super.callTrailer(message, context);
30: }
31: return super.callTrailer(message, context);
32: }
33: }
34: }

Listing 2: New Transformation Module

Compile the code above using the java libraries from the RIM/WebRoot/WEB-INF/lib/ directory and place the compiled library into the same directory.

5.12 Configuring the Registry

The Registry is configured through the editing an XML document. The following table describes the XML elements in the configuration document and provides example values for each element.

Table 3. Registry Configuration Values

<table>
<thead>
<tr>
<th>Name</th>
<th>XML Path</th>
<th>Value Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registry Identifier</td>
<td>/registry/identifier</td>
<td>10233/registry</td>
<td>The registry identifier (a handle), used in all service interactions with the registry. Change the identifier as per your prefix.</td>
</tr>
<tr>
<td>XML Schema Instance</td>
<td>/registry/configuration/schemas/xmlschemainstance</td>
<td><a href="http://www.w3.org/2001/XMLSchema-instance">http://www.w3.org/2001/XMLSchema-instance</a></td>
<td>This is a constant. Use the value as specified.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td><strong>XML Path</strong></td>
<td><strong>Value Example</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Schema Location</td>
<td>$schmloc$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registry Submission Schema</td>
<td>/registry/configuration/schemas/registrysubmissionschema</td>
<td><a href="http://hdl.cordra.net/2000.2.1/ADL-R-Reg-T">http://hdl.cordra.net/2000.2.1/ADL-R-Reg-T</a></td>
<td>This is the submission schema namespace. If there is any change in the LOM schema from the ADL-R LOM schema, use a different namespace. Otherwise, use as specified.</td>
</tr>
<tr>
<td>Registry Submission Schema Location</td>
<td>/registry/configuration/schemas/registrysubmissionsch</td>
<td><a href="http://hdl.cordra.net/2000.2.1/ADL-R-Reg-T?VData=version&amp;id=2">http://hdl.cordra.net/2000.2.1/ADL-R-Reg-T?VData=version&amp;id=2</a></td>
<td>This is the location of the submission schema. Use as specified unless a modified LOM schema is used, in which case, use the new location of the schema.</td>
</tr>
<tr>
<td>Registry LOM Schema Location</td>
<td>/registry/configuration/schemas/registrylomsch</td>
<td><a href="http://hdl.cordra.net/2000.2/adlreg-lom?VData=version&amp;id=2">http://hdl.cordra.net/2000.2/adlreg-lom?VData=version&amp;id=2</a></td>
<td>The ADL-R version of LOM is stated in the example. If a different version of LOM is used, specify the new location here. Otherwise, use as per the example.</td>
</tr>
<tr>
<td>Core Registry URI</td>
<td>/registry/configuration/access/uri</td>
<td><a href="http://doregistry.cnri.net:8080/CORDRAWeb/processrequest">http://doregistry.cnri.net:8080/CORDRAWeb/processrequest</a></td>
<td>The core registry access URI (URL). This depends on the final deployment of the CORDRAWeb module. Replace the DNS name and port from the specified example with the deployment details. IP address may be used instead of DNS names.</td>
</tr>
<tr>
<td>Portal URI</td>
<td>/registry/configuration/access/portaluri</td>
<td><a href="http://doregistry.cnri.net/CORDRAWeb/processrequest">http://doregistry.cnri.net/CORDRAWeb/processrequest</a></td>
<td>Some features are presently being tested for which this detail is necessary. Use the URI as above.</td>
</tr>
<tr>
<td>Name</td>
<td>XML Path</td>
<td>Value Example</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Registry Access</td>
<td>/registry/configuration/access/protocol</td>
<td>http</td>
<td>The core registry access protocol. Only HTTP protocol is currently enabled and tested.</td>
</tr>
<tr>
<td>Protocol</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registry Character</td>
<td>/registry/configuration/kernel/encoding</td>
<td>UTF-8</td>
<td>The registry encoding for content storage and generation. Use as specified.</td>
</tr>
<tr>
<td>Encoding</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registry Version</td>
<td>/registry/configuration/kernel/versions/registry</td>
<td>1.6.1</td>
<td>The version of ADL-R specifications to which the registry is conforming. Future versions of the Registry will flip this around, with ADL-R configured according to the relevant version of the general DO Registry, but for now use as specified in the example.</td>
</tr>
<tr>
<td>Registry Transac-</td>
<td>/registry/configuration/kernel/versions/transaction</td>
<td>1.6.1</td>
<td>The transaction processing module version. The version of the registry business rules validation module. Use as specified in the example. See the Description at the Registry Version entry above for more detail.</td>
</tr>
<tr>
<td>tion Version</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registry Process</td>
<td>/registry/configuration/kernel/versions/process</td>
<td>1.6.1</td>
<td>Use as specified in the example. See the Description at the Registry Version entry above for more detail.</td>
</tr>
<tr>
<td>Version</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search Engine</td>
<td>/registry/configuration/kernel/versions/searchEngine</td>
<td>1.6.1</td>
<td>The catalogue engine version. Use as specified in the example. See the Description at the Registry Version entry above for more detail.</td>
</tr>
<tr>
<td>Version</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>XML Path</td>
<td>Value Example</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-----------------------------------------------</td>
<td>------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Handle Administrator Identifier</td>
<td>/registry/configuration/kernel/admin/handle</td>
<td>0.NA/3333</td>
<td>The identifier (handle) of the handle administrator. This depends on the Handle server configuration.</td>
</tr>
<tr>
<td>Handle Administrator Private Key</td>
<td>/registry/configuration/kernel/admin/privatekey</td>
<td>/Registry/key3333/admpriv.bin</td>
<td>The private key associated with the administrator.</td>
</tr>
<tr>
<td>Handle Administrator Passphrase</td>
<td>/registry/configuration/kernel/admin/passphrase</td>
<td>1212121</td>
<td>The passphrase associated with the private key.</td>
</tr>
<tr>
<td>Handle Administrator Public Key Offset</td>
<td>/registry/configuration/kernel/admin/offset</td>
<td>300</td>
<td>The offset at which the public key is present in the administrator handle. Usually, it is 300.</td>
</tr>
<tr>
<td>Registry Administrator Email Address</td>
<td>/registry/configuration/kernel/admin/email</td>
<td><a href="mailto:gmanepal-li@cnri.reston.va.us">gmanepal-li@cnri.reston.va.us</a></td>
<td>The email address of the registry administrator.</td>
</tr>
<tr>
<td>Registry Mirrors</td>
<td>/registry/configuration/kernel/mirrors/mirror</td>
<td></td>
<td>Any mirrors that the registry recognizes as valid mirrors of itself so that it can propagate the operations to them. Leave blank if none exist. Contact CNRI to enable mirroring.</td>
</tr>
<tr>
<td>Registry Activity Notification Email Subject</td>
<td>/registry/configuration/kernel/watcheremails/subject</td>
<td>ADL-R v1.6.1 Test bed</td>
<td>The email subject of any registration activity notification.</td>
</tr>
<tr>
<td>Registry Activity Notification Email Addresses</td>
<td>/registry/configuration/kernel/watcheremails/email</td>
<td><a href="mailto:gmanepal-li@cnri.reston.va.us">gmanepal-li@cnri.reston.va.us</a></td>
<td>The email address to notify any registration activity. Use this element as many times as needed to add multiple email addresses.</td>
</tr>
<tr>
<td>Name</td>
<td>XML Path</td>
<td>Value Example</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Registry Content Repository Prefixes</td>
<td>/registry/configuration/kernel/contentrepositoriesadministered/prefix</td>
<td>3333</td>
<td>The handle prefix of the content repository that is managed by the registry. If the registry manages a prefix, the content objects handles are created/modified by the registry based on registration. Use this element multiple times to add multiple prefixes. See example for syntax.</td>
</tr>
<tr>
<td>Unregistered Metadata Instance Tombstone Notice Handle</td>
<td>/registry/configuration/kernel/withdrawnhandle</td>
<td>3333/withdrawn</td>
<td>The handle to which the content objects or metadata instances are resolved to when the item is deleted/unregistered. Create a new handle with the URL type pointing to any tombstone screen and specify that handle here.</td>
</tr>
<tr>
<td>Registry Fully Operational Flag</td>
<td>/registry/configuration/kernel/fullyoperational</td>
<td>true</td>
<td>Values are true or false. This option indicates whether or not the registry is operational, i.e., is it accepting user submissions.</td>
</tr>
<tr>
<td>Primary Registry Flag</td>
<td>/registry/configuration/kernel/primary</td>
<td>true</td>
<td>Values are true or false. This indicates whether this registry is a primary or mirror.</td>
</tr>
<tr>
<td>Registry of Registries Flag</td>
<td>/registry/configuration/kernel/roor</td>
<td>false</td>
<td>This option indicates if the registry is a Registry of Registries. At this point in Registry development, it should always be false.</td>
</tr>
<tr>
<td>Registry SMTP Host</td>
<td>/registry/configuration/kernel/smtphost</td>
<td>129.24.24.110</td>
<td>The IP address of the SMTP for sending email messages as and when required.</td>
</tr>
<tr>
<td>Name</td>
<td>XML Path</td>
<td>Value Example</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>----------------------------------------------------</td>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Registry Internal Repository</td>
<td>/registry/configuration/repositor y/identifier</td>
<td>3333/repository</td>
<td>The identifier of the internal repository used to store the metadata.</td>
</tr>
<tr>
<td>Identifier</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registry Internal Repository</td>
<td>/registry/configuration/repositor y/implementationclass</td>
<td>net.cnri.registry.DORepositor y</td>
<td>The repository driver. Use as specified.</td>
</tr>
<tr>
<td>Driver</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registry Signed-up Groups</td>
<td>/registry/configuration/accesscontrolsystem/groups/group/identifier</td>
<td>4444.grp/testgroup</td>
<td>Repeat the parent element ‘group’ as many times as required to configure the group access control. This element is used to describe the group identifier for which the privileges are configured.</td>
</tr>
<tr>
<td>Group Managed Content Repository</td>
<td>/registry/configuration/accesscontrolsystem/groups/group/contentrepository</td>
<td>3333</td>
<td>The privileges for a group are exercised on a content repository with a prefix as described here.</td>
</tr>
<tr>
<td>Registry Index Engine URI</td>
<td>/registry/configuration/catalogueengine/indexuri</td>
<td><a href="http://doregistry.cnri.net:8080/IndexEngine/IndexEngineProxy">http://doregistry.cnri.net:8080/IndexEngine/IndexEngineProxy</a></td>
<td>The core registry index engine URI (URL). This depends on the final deployment of the Catalogue engine module. Replace the DNS name and port from the specified example with the deployment details. IP address may be used instead of DNS names.</td>
</tr>
<tr>
<td>Name</td>
<td>XML Path</td>
<td>Value Example</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>----------------------------------------------------</td>
<td>-----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Registry Search Engine URI</td>
<td>/registry/configuration/catalogueengine/searchuri</td>
<td><a href="http://doregistry.cnri.net:8080/IndexEngine/IndexEngineProxy">http://doregistry.cnri.net:8080/IndexEngine/IndexEngineProxy</a></td>
<td>The core registry search engine URI (URL). This depends on the final deployment of the Catalogue engine module. Replace the DNS name and port from the specified example with the deployment details. IP address may be used instead of DNS names. Usually, it is same as index engine URI.</td>
</tr>
<tr>
<td>Registry Indexing Rules</td>
<td>/registry/configuration/catalogueengine/indexrules</td>
<td>/Registry/conf/index/schemarules</td>
<td>Specify the absolute location of the schemarules file. This file is present under IndexEngine/WebRoot/STATIC-INF directory. Change the indexing rules as and if necessary.</td>
</tr>
<tr>
<td>Registry Indexing Rules for Envelope</td>
<td>/registry/configuration/catalogueengine/enveloperules</td>
<td>/Registry/conf/index/envelopeschemarules</td>
<td>Specify the absolute location of the envelopeschemarules file. This file is present under IndexEngine/WebRoot/STATIC-INF directory. Change the rules as and if necessary.</td>
</tr>
<tr>
<td>Registry Index Directory</td>
<td>/registry/configuration/catalogueengine/indexsegmentsdirectory</td>
<td>/Registry/index/lucene.index</td>
<td>Specify the absolute location of a directory where the index engine stores the indexes.</td>
</tr>
<tr>
<td>Registry Catalogue Engine Log Directory</td>
<td>/registry/configuration/catalogueengine/logdirectory</td>
<td>/Registry/IndexLog</td>
<td>Specify the absolute location of a directory where the index logs are to be stored.</td>
</tr>
<tr>
<td>Authenticate User Flag</td>
<td>/registry/configuration/authenticationsystem/authenticateuser</td>
<td>true</td>
<td>Values are true or false. If the registry must authenticate users, use true, otherwise use false.</td>
</tr>
<tr>
<td>Name</td>
<td>XML Path</td>
<td>Value Example</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------------------------------------------</td>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Authenticate Machine Flag</td>
<td>/registry/configuration/authenticationsystem/authenticateaccess</td>
<td>true</td>
<td>Values are true or false. If the registry must authenticate the IP address of the machine use true, otherwise use false. Contact CNRI if access authentication is needed.</td>
</tr>
<tr>
<td>User Prefix</td>
<td>/registry/configuration/authenticationsystem/userprefix</td>
<td>3333.usr</td>
<td>The handle prefix of the user identifiers. This assumes that all user handles are created under a specific prefix. The end users only use the suffix of the handle when authenticating.</td>
</tr>
<tr>
<td>Validate Submission Flag</td>
<td>/registry/configuration/validationsystem/validate</td>
<td>true</td>
<td>Values are true of false. True, if the submission is to be validated, otherwise false.</td>
</tr>
<tr>
<td>Validation Driver</td>
<td>/registry/configuration/validationsystem/implementationclass</td>
<td>net.cnri.adlr.rim.serviceimpl.preprocess.BusinessLogicValidator</td>
<td>The validation module driver. Unless a custom module was written for validation, use as specified.</td>
</tr>
<tr>
<td>Validation Synchronization Threshold</td>
<td>/registry/configuration/validationsystem/synchronizationthreshold</td>
<td>10485760</td>
<td>The maximum size of the uploaded file before it requires an offline validation. This depends on your bandwidth and load. The present configured value is stated in the example.</td>
</tr>
<tr>
<td>Response Transform Flag</td>
<td>/registry/configuration/transformationsystem/transform</td>
<td>true</td>
<td>Values are true of false. True, if response transformation is needed, otherwise false.</td>
</tr>
<tr>
<td>Transformation Driver</td>
<td>/registry/configuration/transformationsystem/implementationclass</td>
<td>net.cnri.adlr.rim.serviceimpl.postprocess.ResponseTransformer</td>
<td>If a custom response is needed, implement a custom transformation module and use that module as a driver. Otherwise use the one specified in the example.</td>
</tr>
<tr>
<td>Name</td>
<td>XML Path</td>
<td>Value Example</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------</td>
<td>----------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RIM URI</td>
<td>/registry/configuration/rim/uri</td>
<td><a href="https://doregistry.cnri.net/ADLRIM">https://doregistry.cnri.net/ADLRIM</a></td>
<td>The RIM URI. The URI must be absolute and must include the servlet (or any related name) before the query parameters begin. For example, if <a href="http://registry:8080/ADLRIM/gateway?registry=4444/registry&amp;">http://registry:8080/ADLRIM/gateway?registry=4444/registry&amp;</a>..... is used for one of the RIM interfaces, the value here must be <a href="http://registry:8080/ADLRIM/gateway">http://registry:8080/ADLRIM/gateway</a>.</td>
</tr>
<tr>
<td>RIM Version</td>
<td>/registry/configuration/rim/version</td>
<td>1.0</td>
<td>Currently 1.0</td>
</tr>
<tr>
<td>RIM Access Protocol</td>
<td>/registry/configuration/rim/accessprotocol</td>
<td>http</td>
<td>HTTP is the supported protocol in this version</td>
</tr>
<tr>
<td>Error Messages URI</td>
<td>/registry/configuration/rim/errormessageuri</td>
<td><a href="https://doregistry.cnri.net/ADLRIM/errorcodes.html">https://doregistry.cnri.net/ADLRIM/errorcodes.html</a></td>
<td>It should point to the error codes html file. In the present distribution, the html file can be found at http://&lt;RIMURI&gt;/errorcodes.html</td>
</tr>
<tr>
<td>RIM Error Message Schema Namespace</td>
<td>/registry/configuration/rim/responseschema</td>
<td><a href="http://hdl.cordra.net/2000.2.1/rim-error-status">http://hdl.cordra.net/2000.2.1/rim-error-status</a></td>
<td>This is the RIM response schema namespace. Use as specified.</td>
</tr>
<tr>
<td>RIM Error Message Schema Location</td>
<td>/registry/configuration/rim/responseschemalocation</td>
<td><a href="http://hdl.cordra.net/2000.2.1/rim-error-status?VData=version&amp;id=1">http://hdl.cordra.net/2000.2.1/rim-error-status?VData=version&amp;id=1</a></td>
<td>This is the location of the RIM response schema. Use as specified.</td>
</tr>
<tr>
<td>RIM Metrics Schema</td>
<td>/registry/configuration/rim/metricschema</td>
<td><a href="http://hdl.cordra.net/2000.2.1/rim-metrics-response">http://hdl.cordra.net/2000.2.1/rim-metrics-response</a></td>
<td>This is RIM metrics schema namespace. Use as specified.</td>
</tr>
<tr>
<td>RIM Metrics Schema Location</td>
<td>/registry/configuration/rim/metricschemalocation</td>
<td><a href="http://hdl.cordra.net/2000.2.1/rim-metrics-response?VData=version&amp;id=1">http://hdl.cordra.net/2000.2.1/rim-metrics-response?VData=version&amp;id=1</a></td>
<td>This is the location of the RIM metrics schema. Use as specified.</td>
</tr>
<tr>
<td>Name</td>
<td>XML Path</td>
<td>Value Example</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>----------------------------------------------------</td>
<td>----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Validation Transformation Document</td>
<td>/registry/configuration/rim/validationtransformationdocument</td>
<td>/Registry/rim/STATIC-INF/adl_reg_sch1-5.xsl</td>
<td>Specify the absolute location of the adl_reg_sch1-5.xsl file. This file is presently under RIM/WebRoot/STATIC-INF directory.</td>
</tr>
<tr>
<td>Validation Transformation Skeleton Document</td>
<td>/registry/configuration/rim/validationtransformationSkeletonDocument</td>
<td>/Registry/rim/STATIC-INF/skeleton1-5.xsl</td>
<td>Specify the absolute location of the skeleton1-5.xsl file. This file is presently under RIM/WebRoot/STATIC-INF directory.</td>
</tr>
<tr>
<td>Validation Document</td>
<td>/registry/configuration/rim/validationdocument</td>
<td>/Registry/rim/STATIC-INF/schematron.xml</td>
<td>Specify the absolute location of the schematron.xml file. This file is presently under RIM/WebRoot/STATIC-INF directory.</td>
</tr>
<tr>
<td>Virus Scan Flag</td>
<td>/registry/configuration/rim/virusscan</td>
<td>false</td>
<td>Specify if a virus scan is needed for every submission made to the registry. Values are true or false. If true, make sure there is a command line virus scanner installed on the machine. The executable must be present in a particular location. Refer to the corresponding section in this manual for full details on this.</td>
</tr>
<tr>
<td>Submission Response Type</td>
<td>/registry/configuration/rim/servicetypes/submissionresponsetype</td>
<td>2000.2.3/1</td>
<td>This is a constant.</td>
</tr>
<tr>
<td>Search Response Type</td>
<td>/registry/configuration/rim/servicetypes/searchresponsetype</td>
<td>2000.2.3/2</td>
<td>This is a constant.</td>
</tr>
<tr>
<td>Transaction Status Retrieval Response Type</td>
<td>/registry/configuration/rim/servicetypes/transactionstatusretrievalresponsetype</td>
<td>2000.2.3/1</td>
<td>This is a constant.</td>
</tr>
<tr>
<td>Name</td>
<td>XML Path</td>
<td>Value Example</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Validation Status Retrieval Response Type</td>
<td>/registry/configuration/rim/service-types/validationstatusretrievalresponse</td>
<td>2000.2.3/1</td>
<td>This is a constant.</td>
</tr>
<tr>
<td>Metadata Retrieval Response Type</td>
<td>/registry/configuration/rim/service-types/metadataretrievalresponse</td>
<td>2000.2.3/3</td>
<td>This is a constant.</td>
</tr>
<tr>
<td>Metadata Instance Information Retrieval Response Type</td>
<td>/registry/configuration/rim/service-types/metadatainstanceinformationretrievalresponse</td>
<td>2000.2.3/4</td>
<td>This is a constant.</td>
</tr>
<tr>
<td>Metrics Evaluation Response Type</td>
<td>/registry/configuration/rim/service-types/metricsevaluationresponse</td>
<td>2000.2.3/5</td>
<td>This is a constant.</td>
</tr>
<tr>
<td>SQL Query Response Type</td>
<td>/registry/configuration/rim/service-types/sqlqueryresponse</td>
<td>2000.2.3/6</td>
<td>This is a constant.</td>
</tr>
<tr>
<td>SQL Update Response Type</td>
<td>/registry/configuration/rim/service-types/sqlupdateresponse</td>
<td>2000.2.3/7</td>
<td>This is a constant.</td>
</tr>
<tr>
<td>CORE Retrieval Response Type</td>
<td>/registry/configuration/rim/service-types/coreretrievalresponse</td>
<td>2000.2.3/6</td>
<td>This is a constant.</td>
</tr>
<tr>
<td>Feed Response Type</td>
<td>/registry/configuration/rim/service-types/feedresponse</td>
<td>2000.2.3/7</td>
<td>This is a constant.</td>
</tr>
<tr>
<td>Database Support Flag</td>
<td>/registry/configuration/rim/database/suppose</td>
<td>true</td>
<td>Specify if database support through RIM is needed or not. Values are true or false.</td>
</tr>
<tr>
<td>Database URI</td>
<td>/registry/configuration/rim/database/uri</td>
<td>jdbc:mysql://localhost:3306/rimlitedb</td>
<td>The database jdbc URI. The value in the example shows the pattern. This DB schema is irrelevant to RIM. RIM just provides a SQL interface via RIM to the underlying DB.</td>
</tr>
<tr>
<td>Database Username</td>
<td>/registry/configuration/rim/database/username</td>
<td>rimuser</td>
<td>The database username.</td>
</tr>
<tr>
<td>Database Password</td>
<td>/registry/configuration/rim/database/password</td>
<td>rimpasswd12</td>
<td>The database password.</td>
</tr>
<tr>
<td>Name</td>
<td>XML Path</td>
<td>Value Example</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------</td>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Database JDBC Driver</td>
<td>/registry/configuration/rim/database/driver</td>
<td>com.mysql.jdbc.Driver</td>
<td>The name of the driver. If MySQL is used as a DBMS, use as specified. Otherwise, refer to appropriate vendor manual.</td>
</tr>
</tbody>
</table>

Modify the configuration XML document using the above table.

### 5.13 Update HTML Files with your Registry Handle

There are a few HTML files that need to be edited so that Core Registry test pages will display the correct handle value in the “Target Registry” input field of each page. While this step is not essential for the operation of the Registry Software, these pages help test and validate the Registry Installation.

These files are located in directory `<adl_home>/CDOR/Kernel/WebRoot/`:

- SubmitCORDRABatch.html
- SearchRequest.html
- RequestBatchStatus.html
- RequestValidationStatus.html

Similarly for the files located in directory `<adl_home>/CDOR/RIM/WebRoot/`:

- SubmitCORDRABatch.html
- SearchRequest.html
- RequestBatchStatus.html
- RequestValidationStatus.html
- MetricsRequest.html
- SQLQuery.html
- SQLUpdate.html

Each file must be edited to point to the registry that is being tested. For example, we installed and tested the registry at 10233/registry. We edited these HTML files to enable the user to select the Registry at 10233/registry as shown:
Once the files have been updated, the configure batch file must be run. Our example run is shown in Figure 11 below:

![Configure Batch File](image.png)

Figure 11. Configure Batch File
5.14 Installing CNRI Repository Software

Installing the Registry consists of the following tasks:

- Deploying Core Registry
- Deploying Index Engine
- Deploying RIM-LITE

5.14.1 Deploying Core Registry

Deploy the CORDRAWeb.war to the J2EE Web Container Server. Please refer to the documentation for your Web Container as this step is specific to the deployment process for your Web Container.

For our example, using Tomcat, we deploy as shown below:

```
adl@/home/adl/CDOR>cp ./CORDRAWeb.war /usr/local/registry/apache-tomcat-5.5.25/webapps
```

5.14.2 Deploying Index Engine

Deploy the IndexEngine.war to the J2EE Web Container Server.

For our example, using Tomcat, we deploy as shown below:

```
adl@/home/adl/CDOR>cp ./IndexEngine.war /usr/local/registry/apache-tomcat-5.5.25/webapps
```

5.14.3 Deploying RIM-LITE

Deploy the ADLRIM.war to the J2EE Web Container Server.

For our example, using Tomcat, we deploy as shown below:

```
adl@/home/adl/CDOR>cp ./RIM.war /usr/local/registry/apache-tomcat-5.5.25/webapps/ADLRIM.war
```

In our case, we modified the RIM.war to ADLRIM.war. The war files may be renamed at will, as long as the URLs are properly configured in the configuration XML document.
6 Registry Startup and Shutdown

The Registry Startup requires you to perform the following steps:

- Startup Handle Server
- Startup the Database
- Startup DO Repository
- Startup Web Container(s) which run the deployed components
  - Core Registry
  - Index Engine
  - RIM-LITE

We recommend that you follow the instructions specific to your operating system to automate startup, and shutdown the processes as ‘services’. Below we show how to startup and shutdown using the command line.

6.1.1 Handle Server Startup and Shutdown

For our example, the Handle Server is started as shown below:

```
% java -cp /home/adl/hdl6.2.5_01/bin/handle.jar net.handle.server.Main /home/adl/hdl6.2.5_01/svr_1
```

We shut down the Handle Server using ctrl-c from the command line or by using the kill command. This shuts down the Handle Server gracefully.

Note: Do not use SIGKILL signal to kill the Handle Server as that signal is not trapped and the process will not terminate gracefully.

6.1.2 Database Startup and Shutdown

The Database startup and shutdown is specific to the Database system you are using.

6.1.3 DO Repository Startup & Shutdown

For our example, we startup the DO Repository server using the command:
% do-server /usr/local/registry/data/
We shutdown the DO Repository Server using ctrl-c from the command line or using the kill command. This shuts down the Repository Server gracefully.

**Note:** Do not use SIGKILL signal to kill the Repository Server as that signal is not trapped and the process will not terminate gracefully.

### 6.1.4 Web Container Startup and Shutdown

Please follow the steps specific to your Web Container / Application Server to startup and shutdown.

As an example, we startup Tomcat using the command:

```
adl@/usr/local/registry/apache-tomcat-5.5.25/bin>startup.sh
```

Tomcat is shutdown using the command:

```
adl@/usr/local/registry/apache-tomcat-5.5.25/bin>shutdown.sh
```

### 7 Post Installation Validation

#### 7.1 Testing the Core Registry

Basic validation of the Core Registry installation can be done using a browser based UI. The following operations can be tested using this UI:

- Batch Submission
- Request Batch Submission Status
- Request Validation Status
- Search Registry

The Web UI can be accessed at `http://<host:port>/CORDRAWeb/`. For our example, the Core Registry Web UI can be accessed at `http://63.88.245.55:8080/CORDRAWeb/`, as shown below.
At this stage, exercise all the available functions on this web page, and ensure that they execute successfully. For example, the Batch Submission can be tested as shown below:
7.2 Testing the Registry using RIM-LITE

Every client connecting to RIM-LITE should first obtain an active Client Access Key (CAK). The CAK is tied to the IP of the machine where the client executes. The CAK itself is a handle. The step that completes the post-installation validation using RIM-LITE requires a CAK. If you don’t currently have a CAK, you can obtain one by contacting support via the information provided in the ‘Support’ section.
For example, we tested RIM-LITE using the browser running on a machine with IP 156.80.172.46. The CAK assigned to us was 2000.2.4/27.

The entire Registry installation, including RIM-LITE, can be validated by accessing each of the functions available at http:<host:port>/ADLRIM/.

For example, we accessed our installation at http://63.88.245.55:8080/ADLRIM/, as shown below.
Figure 14. Validating the Complete Registry Installation

As an example, the ‘Submit Batch’ functionality can be tested as shown below. Note the CAK field on the form.
Figure 15. Example of Validating the Registry Batch Submission Function
8  Appendix A – Overview of the Handle Admin Tool

This section provides a quick overview of the functionality of the Handle Admin Tool. The Handle Admin Tool is a part of the HANDLE.NET Software and is available at http://www.handle.net.

A handle can be administered using the Handle Admin Tool.

Note: Handle are case-sensitive. So a handle of value “0.NA/10233” is different from the value of “0.na/10233”. Pay special attention to using the proper case, with the correct usage being “0.NA”. Several installation problems can be traced back to improper use of case.

8.1 Starting the Handle Admin Tool

Start the Handle Admin Tool using the directory path names specific to your installation. For example, we ran the Handle Admin Tool from a Windows machine using the following command:

```
java -cp c:\hd16.2.5_01\handlet.jar net.handle.apps.gui.hadmin.HandleTool
```

This starts the Handle Admin Tool as shown below:
8.2 Authenticating to the Handle System

To create or modify a handle, you first need to authenticate by going to Setup -> Authentication.
In this example, we are authenticating as the Handle Server Administrator for prefix ‘10233’, using public/private key authentication.
Figure 18. Handle System Authentication – Step 2

Notice the change of display at the bottom of the Tool Main Window to display the ‘authentication’ information:
8.3 Creating or Modifying a Handle

A handle is comprised of several data values. Each data value in a handle has a unique index. The value of the index needs to be unique across all data values for that handle. While the actual value of the index is not important, some commonly used conventions are:

- HS_ADMIN values start with the index value of 100
- Admin Groups start with the value of 200
The public key index is 300

For complete information on the Handle System, please go to http://www.handle.net.

Please see below for some common examples of providing data values to setup handles.

Add Handle Administrator:

![Add Handle Administrator window](image)

Figure 20. Add Administrator to Handle
Add URL:

![Add URL to Handle](image1)

**Figure 11. Add URL to Handle**

Add Email:

![Add Email to Handle](image2)

**Figure 22. Add Email to Handle**

Once all the data values are set up, click ‘Create’ button on Handle Creation screen to create the handle. If the handle was created successfully, you’ll get a response similar to the one below:

![Successful Handle Creation](image3)

**Figure 23. Successful Handle Creation**