



# RepoMMan Project

---

D-D11

Investigating methods to access user profile data

Robert Sherratt and Richard Green

October 2006

version 1.1 January 2007 [First implementation with RepoMMan tool]



## The RepoMMan Project

<b>Project Director:</b>	Ian Dolphin, Head of e-Strategy, University of Hull (i.dolphin@hull.ac.uk)
<b>Project Manager:</b>	Richard Green (r.green@hull.ac.uk)
<b>Technical Lead:</b>	Robert Sherratt (r.sherratt@hull.ac.uk)
<b>Repository Domain Specialist:</b>	Chris Awre (c.awre@hull.ac.uk)

The Repository Metadata and Management Project (RepoMMan) at the University of Hull is funded by the JISC Digital Repositories Programme. The project is being carried out by the University's e-Services Integration Group (e-SIG) within Academic Services.

## Introduction

The RepoMMan project is researching and developing a workflow tool to facilitate the use of a Fedora-based Institutional Repository. Part of the tool's functionality is to be the ability to populate, or at least pre-populate, metadata for the digital objects created with it. An important element of that metadata relates to the author of the content that the digital object will contain. It is intended that the version of the RepoMMan tool used at the University of Hull will be surfaced in either the University's portal (uPortal), or else in a collaboration and learning environment (C&LE), probably Sakai.

This report details investigative work into methods to access user profile data from the portal framework and a potential enterprise directory such that it can be fed into object metadata by the RepoMMan tool.

## Accessing profile data from uPortal

Since 2003 the University of Hull portal has been deployed using the open source uPortal framework from the JA-SIG consortium. Originally starting with version 2.1.4, the framework was upgraded over the summer of 2006 to use a later version, 2.5.2. This version has, using Distributed Layout Manager, introduced more flexible management of what users see when they login and it is conformant with the widely adopted JSR-168 portlet standard.

To understand how basic profile metadata can be accessed from uPortal, it is first necessary to have an overview of what happens when a user logs in to the portal. Once a user has entered their credentials into the authentication form a login servlet is called and the user's credentials are passed as part of the request. These credentials are then forwarded to an LDAP authentication service. If the user is successful in authenticating, the next step is to populate their profile.

How the portal profile is populated depends on the configuration of the person directory service in the PersonDirs.xml file. The person directory service can either extract profile data from an enterprise directory or, as currently used at Hull, a relational database. The service is configured with a SQL statement to select appropriate data with the user-id as the unique key. This statement selects data from a central identity database. Below are the first few lines of the query:

```
SELECT frnm + ' ' + srnm as FIRST_LAST,
       frnm as FIRST_NAME,
       srnm as LAST_NAME,
       email as EMAIL ,
```

Once the data has been retrieved, individual pieces of data are then mapped to attributes. The profile is based on the eduPerson standard, extended for some specific requirements at Hull. Below is the syntax for mapping a profile attribute to data retrieved by the person directory service.

```
<attribute>
  <name>FIRST_LAST</name>
  <alias>displayName</alias>
</attribute>
```

Once the profile data has been retrieved this is used to populate a Java object, IPerson, that persists for the duration of the user's portal session. The IPerson object is available to any class, servlet or JSP within the uPortal application context. Any of those applications can retrieve data from the IPerson object using code similar to that shown below:

```
IPerson person =  
PersonManagerFactory.getPersonManagerInstance().getPerson(request);  
String username = (String) person.getAttribute("userName");
```

In the case of portlets, this information can automatically be passed to any method that will call the repository.

Note that this method is now being implemented in the alpha version of the RepoMMan tool. In this latest instance of the tool interface, an Adobe Flex application distributed as a Flash executable, a JSP calls the IPerson object, retrieves the required data and passes it to the Flash application which is embedded in the JSP.

## Accessing profile data from an enterprise directory

At the time of writing, the University of Hull does not have an enterprise directory as such. The University network makes use of an LDAP directory for basic authentication; user attributes for the purpose of authorisation and the like are drawn from a range of other on-line systems as required. Proof of concept work is ongoing with Sun Microsystems with a view to possible introduction of an enterprise directory based on Sun Directory Server 5.2 and using LDAP. The main thrust of this investigation is to establish whether an enterprise directory can support the RepoMMan tool, however this is related to the larger question of whether an LDAP server can support Fedora. Only if Fedora will operate in this way will it be sensible to use the same LDAP server for RepoMMan.

The current production version of Fedora 2.1.1 natively uses a Tomcat user file to store details of users, passwords and their permitted roles within the Fedora repository. The role information can be used as the basis for complex and sophisticated security arrangements implemented through the Fedora XACML engine. Fedora, as shipped, does not use LDAP for primary authentication but can use an LDAP directory as an extension of the Tomcat authentication process. Once a user has successfully authenticated against Tomcat, Fedora will pass the credentials to an LDAP server to determine whether it holds additional role information for the user and, if so, retrieve it. There is no 'official' method provided with Fedora 2.1.1 to make an LDAP server the primary authentication mechanism although there are a number of pointers as to how it might be done.<sup>1</sup> This situation is presumably predicated on the idea that a conventional repository server has few users who need to interact with the system to deposit and manage materials; the server may, of course, have many thousands of anonymous users who wish to access its contents in a read-only fashion.

At the University of Hull, we intend that our institutional repository should be used to support the production of digital objects as well as to be a storage and access point for the finished materials. Thus we anticipate that we shall have a user base measured in thousands requiring to interact with the repository as well as the many users accessing it as anonymous readers. Clearly, it would not be appropriate to attempt authentication on such a scale using a Tomcat user file. Management of that file, let alone the repository itself, would be a daunting task. Accordingly it was determined that we should attempt to configure Fedora and Tomcat such that the authentication mechanism was reversed: LDAP would be the primary authentication mechanism whilst the Tomcat user file - if needed - could act as a secondary provider of role information.

As noted above, the University does not currently operate an LDAP server containing both authentication and role information. Accordingly it was decided to use for testing the small LDAP server implementation used for the Sun proof of concept work. This became available in the early autumn of 2006.

The test LDAP server has a standard structure and we anticipate that a very similar structure will be employed for the enterprise directory: the root of the server contains, amongst others, two organisational units (ou) 'people' and 'groups'. The 'people' section contains an entry for all the users known by the LDAP with basic information about them. A person may optionally be listed as a member of one or more of the groups defined in the 'groups' section in order to

be ascribed associated roles for the purpose of authorisation; these headings might also be useful for metadata.

Thus an entry in the 'people' section might be:

userPassword	(binary encoded)
uid	acsrg
givenName	Richard
objectClass	top
objectClass	person
objectClass	organizationalPerson
objectClass	inetorgperson
sn	Green
cn	Richard A Green

and this user might be represented in one of the groups thus:

objectClass	top
objectClass	groupofuniqueNames
uniqueMember	uid=acsrg,ou=people,dc=hull,dc=ac,dc=uk
cn	staff

and in another thus:

ou	groups
description	Users with UoHPS Fedora role
objectClass	top
objectClass	groupOfUniqueNames
uniqueMember	uid=acsrg, ou=people, dc=hull, dc=ac, dc=uk
cn	fedoraRole=UoHPS

Here is defined a user Richard Green with a user-id 'acsrg' and a password. Thus basic authentication should be possible. Further, he is a member of the group 'staff' which will be used in University systems to determine a particular level of authorisation, and in another group 'fedoraRole=UoHPS' which can theoretically be used by Fedora for a similar purpose. (Note that the use of an '=' sign in the second group name is a specific Fedora construct.)

It should now be possible to use a 'realm element' to search for a user's roles. The example given assumes that a user logs in to Fedora using their user-id and that an anonymous connection to the LDAP server is sufficient to retrieve role information:

```
<Realm className="org.apache.catalina.realm.JNDIRealm" debug=99
  connectionURL="ldap://xxx.hull.ac.uk:389"
  userPattern="uid={0}, ou=people, dc=hull, dc=ac, dc=uk"
  roleBase="ou=groups, dc=hull, dc=ac, dc=uk"
  roleName="cn"
  roleSearch="(uniqueMember={0})"
/>
```

With this configuration the realm should assemble the user's distinguished name (dn) by substituting the user-id into the `userPattern` at `uid={0}`, authenticate with the directory with this distinguished name and the password received from the user, and search the directory to find the user's roles. The example realm element follows closely an example provided by the Apache organisation in its documentation for Tomcat 5.0 the web server under which Fedora 2.1.1 is run.<sup>2</sup>

Our first attempt to have Fedora 2.1.1 communicate with the LDAP server failed and the log files were less than helpful in determining the problem. A number of configuration settings were tried but to no avail. We fell back on writing a PHP script to interact with the LDAP server to check that it was functioning correctly. The script attempted to authenticate with the LDAP server and then to go on and retrieve the basic information about user 'acsrg'. (We acknowledge the use of parts of PHP script examples from [uk.php.net](http://uk.php.net)<sup>3</sup>) The following output was generated:

```
Starting check
version 3
verification on 'ldap://xxx.hull.ac.uk': ACCESS GRANTED authentication= true
1 entries returned
uid: acsrg
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgperson
givenname: Richard
sn: Green
cn: Richard A Green
```

Clearly, the LDAP server is working as it should and authentication in Fedora could be expected.

A fairly involved exchange of e-mails with the Fedora development team ensued. They were not aware of anyone else attempting to use Fedora with LDAP in this way and so were unable to point us at an immediate source of help. Indeed they noted that the company VTLS which uses Fedora as the basis for its repository products had written its own LDAP module for use with Fedora rather than use the functionality provided. In due course, the Fedora team member who had developed the LDAP functionality revisited the code and agreed that the version shipped in Fedora 2.1.1 did not work correctly. The view was that, as Fedora 2.2 will ship with servlet-based authentication, it was not productive to attempt to 'fix' the code in 2.1.1. Our understanding is that Fedora 2.2 will support large-scale authentication when it is released in January 2007.

The test using a PHP script, described above, replicates functionality that is straightforwardly transferable into the RepoMMan tool. The tool itself will not require user role information for authorisation purposes but rather to help populate metadata. It has not yet been decided precisely what information will be held on the University enterprise directory but the RepoMMan team is in a position to influence that decision and hence the data that will eventually be available to the RepoMMan tool. The nature of the information that is desirable for personal metadata is discussed elsewhere.<sup>4</sup>

Prior to the introduction of the enterprise directory it may prove necessary to write a temporary routine or routines to extract basic user information from existing University systems. Equally, during the testing phase of the RepoMMan tool it may be possible to use the test LDAP server referred to above to manage a relatively small number of users.

## **Conclusions**

As regards the use of an enterprise directory, it clearly is feasible to retrieve user profile data from such a system and this will be attempted if and when such a system is available to us. In the interim, we shall use the method detailed in the first section to extract information from the current central identity database.

## References

- 
- <sup>1</sup> Fedora Project (2006) *Securing your Fedora Repository (Fedora 2.1)* p4ff  
at <http://www.fedora.info/download/2.1/userdocs/server/security>
  - <sup>2</sup> Apache Organisation *Tomcat 5.0 documentation*  
at <http://tomcat.apache.org/tomcat-5.0-doc/realms-howto.html#JNDIRealm>
  - <sup>3</sup> UK PHP Net *LDAP functions*  
at <http://uk.php.net/ldap>
  - <sup>4</sup> Green R (2007) *R-D11 Report on metadata needs for the University of Hull Digital Repository* RepoMMan Project, University of Hull  
at <http://www.hull.ac.uk/esig/repomman/documents> (valid January 2007 - RG)