URDMS DMP Application

# Overview

The DMP (Data Management Plan) is an ASP MVC3 web application is the starting point for researchers to complete/update their data management plans, data deposit forms, data collections and some project information. The web application kicks off SharePoint site provisioning by sending an NServiceBus message to the provisioning service on completion of a DMP/DD. Data Collection approvals are also completed via the web application, although the workflow is managed by the approvals service.

# Installation

* Create “Urdms” and “UrdmsTests” MS SQL databases.
* Open the Urdms.Dmp.sln solution file via Visual Studio.
* Change configuration items as required (See Web.Config setting section below).
* Setup URDMS application to run via the local IIS sever (see IIS Setup section below).
* Rebuild the application and run it to allow migrator.NET to create the tables.
* The Urdms Web application relies on the NService Bus queues being in place for the application to function. Please refer to the URDMS Integration documentation.

# Implementation

* Implement ICurtinUserService interface (currently a mock implementation is used – DummyUserLookupService in the DummyService.cs file).
* Implement the IMemberService interface (currently a mock implementation is used – DummyMemberService in the DummyService.cs file). This service adds roles to the user.
* Implement the IDirectoryEntryService interface (currently a mock implementation is used – DummyDirectoryEntryService in the DummyService.cs file). This class is used for extracting user roles via directory service such as Active Directory.
* Add user roles for controlling top navigation menu accessibility (needs to be one of the roles added to the user via the Member or DirectoryEntry service.

# Test run

The DummyService.cs file has mock implementations for the ICurtinUserService, IMemberService and DirectoryEntryService. In addition it also has sample IDs for a researcher and approvers that can be used for testing the applications functionality. To login to the site use any of these IDs as the Login ID and any string in the password.

# Web.config settings

Some of the web configuration items noted below will require transforms to be defined in the CI, Prod and Test environments.

* NServiceBus endpoints for Urdms.ProvisioningService.Commands, Urdms.Approvals.ApprovalService.Commands and Urdms.DocumentBuilderService.Commands message type end point queue must be defined here (the web application endpoint is defined in the AutoFac NServiceBusModule (under Config/Autofac).
* If directory (i.e. Active Directory) based roles are to be used for the application the set values for LdapURI, LdapUser and LdapPassword. These are used as parameters for the for the DirectoryEntryService helper class which is used in UrdmsRoleProvider.
* CsvSeparator is used for the CSV exporter.
* LibGuideSource is used as part of the LibGuideService. While the service and controller logic is in place it is not currently used in the site.

# Setting up IIS (Windows 7)

1. Create a web application via the IIS manager console
   1. Under the machine name in the connections panel right click on “Sites” and select “Add web application”
   2. Add site name as “urdms.local”.
   3. Set the “Physical path” point to the location of the MVC application for URDMS.
   4. Set the host name as “urdms.local”
   5. Click “Test Settings” if the user that site is running under has required access (in some instances you might have to set that user to administrative access to the machine)
   6. Make a note of the name of the application pool
2. Set application pool settings
   1. Select (not already selected) “v4.0” as the “.NET Framework Version”.
   2. Select previously notes application pool name under the “Application Pools” section in connections panel.
   3. Right click on the pool and select “Advanced Settings”.
   4. Change the “enable 32-Bit Applications” to “False”.
   5. You might have to set the Identity to a user who has administrative rights to the machine.
3. Add the following line to the host file located at C:\Windows\System32\drivers\etc\hosts

127.0.0.1 urdms.local

# Logging

Currently Elmah is set to log unhandled exceptions using in-memory storage and presenting a web view at ~/elmah.axd . Elmah is module-based and allows configuration via web.config - production settings will need to be addressed to eliminate security risks.