ICT MyMedia Project
2008-215006

Field Trial Integration Quickstart

17 June 2009

Public Document

1 Contents

2 Target Audience 4

3 Introduction 4

4 How To Implement 4

4.1 Proposed Field Trial Architecture/Deployment 4

4.2 Overview of the WCF Sample Project 6

4.3 Deploy WCF Sample to IIS Web Server 10

Project acronym: MyMedia

Project full title: *Dynamic Personalization of Multimedia*

Work Package:	**2**

Document title:	**Field Trial Integration Quickstart**
Version:	**N/A**
Official delivery date:	**N/A**
Actual publication date:	N/A
Type of document:	Open Source Software and Documentation
Nature:	Public

Authors:	Rich Hanbidge, EMIC
Approved by:	N/A

2 Target Audience

The intended target audience of this document are consumers of the MyMedia project open source software framework.

See the MyMedia softare license, included at the root of the distribution, for terms of use.

3 Introduction

This document describes how to deploy the WCF Sample Application, included in the MyMedia Project source code, into IIS. This provides a starting point for integrating the MyMedia Recommender Framework into a field trial scenario. In addition, this document discusses how to extend the sample application to add more useful features.

4 How To Implement

4.1 Proposed Field Trial Architecture/Deployment

This sections provides guidance for integrating the MyMedia framework into an existing software solution. The deployment assumes a web server, such as IIS, which is capable of supporting WCF (Windows Communication Foundation) based Web Services. These are mere technical suggestions, and are not requirements imposed by the MyMedia framework.

The intent of this design is to provide a separate set of functionality for producing recommendations, which has minimal impact on the operation of the rest of the application. As such, the recommender functionality is exposed through a WCF Web Service interface, allowing a great deal of flexibility, by using open network based protocols.

The recommender framework is devided into roughly 3 components. The recommender component, used for run-time production of recommendations for the users. The second component is the Catalog Importer, which provides a link to the catalog items available for recommendation, as well as bulk update of user training data. Lastly, MDE (Metadata Enrichment), which provides additional information to the catalog for use by the application and algorithms.

We will discuss three models of deployment. The first, is a “development” scenario, in which all components are running in the same process. This presents the most simple scenario, and should provide good development/debugging efficientcy. Secondly, is the “prescribed” model, which logically separates recommendation-time functionality, versus occasionally run functionality. Lastly, we will provide a fully distributed model, in which all of the logical recommendation components run in separate processes.

4.1.1 Prototype/Development Model

This model incorporates all of the components in one process space. This allows for easy debugging of the various components. See Figure 1, Prototype/Development Model Architecture, below.

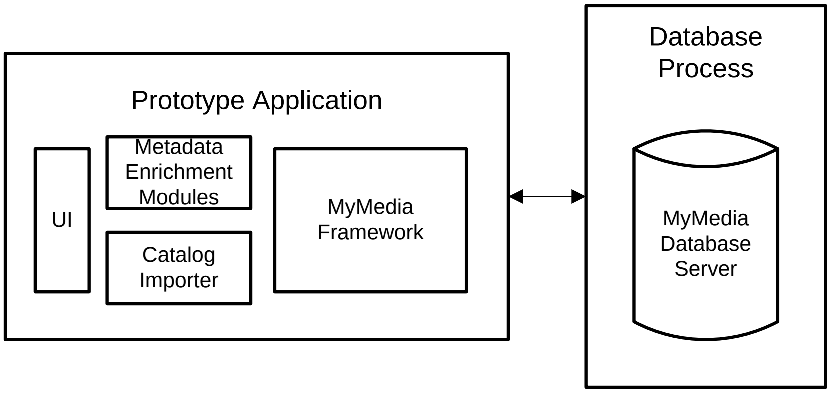


Figure 1 Prototype/Development Model Architecture

4.1.2 Prescribed Deployment Model

This model separates the core components into two process. This separation is between recommendation-time functionality, and occasional updates, such as MDE and Catalog Updates. See Figure 2, Prescribed Deployment Model, below.

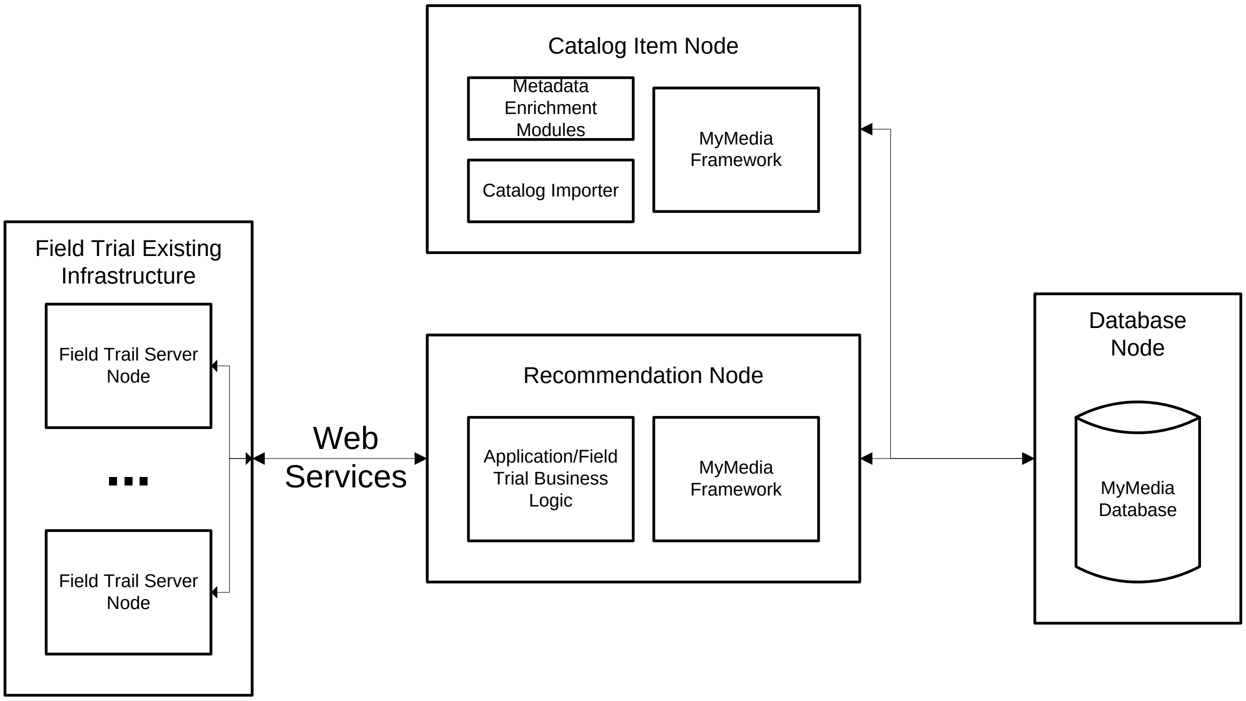


Figure 2 Prescribed Deployment Model

4.1.3 Fully Distributed Model

This model separates all of the logical components of the recommender. See Figure 3, Fully Distributed Model, below.

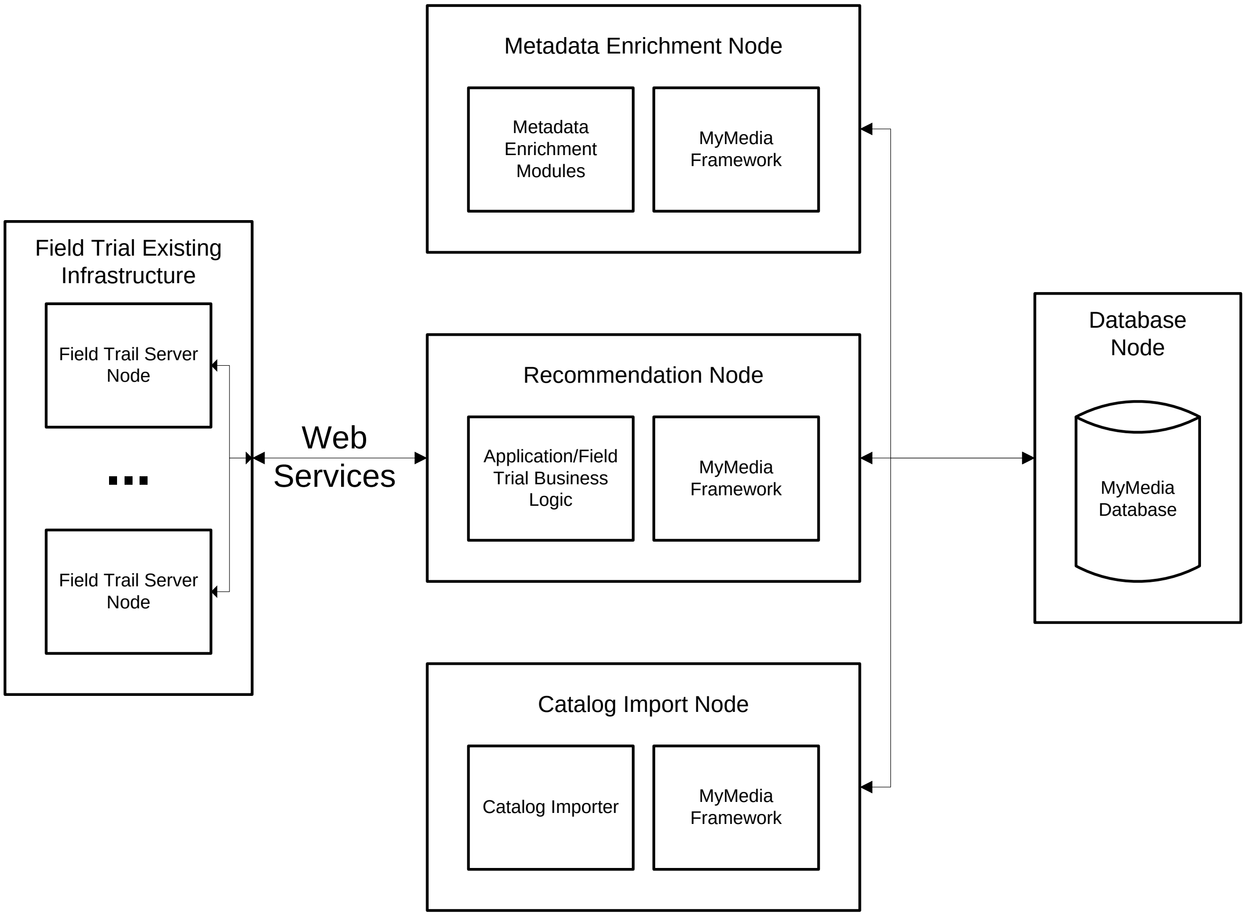


Figure 3 Fully Distributed Model

4.2 Overview of the WCF Sample Project

The WCF sampe is separated into two parts, the client and the server. The server is a WCF Web Service, and the client is a simple winform UI, consuming an automatically generated Web Services client.

4.2.1 Recommender WCF Service

4.2.1.1 Prerequisites

Before running the WCF Service, the MovieLensRecommender project must be run, as this produces the database on which the Recommender WCF service depends. Please see the “MyMedia Framework How To Recommender” document for details and a walkthrough of this.

4.2.1.2 Sample Source Code

The project consists of a SVC file, which defines the service. This service implements one method, SampleRecommenderWebService.GetRatingsForUser. Since this is a web service host, which can support many simultaneous requests, the recommender has been impelemnted as a Singleton, using a wrapper class provided by the MyMedia framework.

The following is a simple walkthrough of the important details of the code.

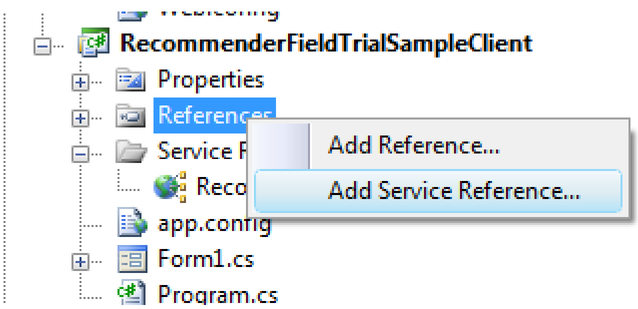
1. SampleRecommenderWebService.GetRatingsForUser, in ...\RecommenderFramework\RecommenderFieldTrialSampleWCF\SampleRecommenderWebService.svc.cs
    a. This class exposes access to the user rating history, via the user id
        i. In a real field trial, the method signature should change from a integer to a string or other object, used to reference the user in the MyMedia framework.
        ii. For example, this may be the user logon string.
    b. An instance of the recommender singleton is retrieved.
    c. The user object is retrieved from the framework using Recommender.GetUser.
    d. The user's feedback is retrieved using the Recommender.GetUserFeedback method.
    e. Lastly, the information from the RatingAction object is converted to the WebService interface object, "UserRating," (see below).
2. UserRating class, defined in ...\RecommenderFramework\RecommenderFieldTrialSampleWCF\UserRating.cs
    a. This class defines the "business logic" to interact between the client/application and the framework. This is determined by the system in which the recommender is being integrated, and it situationally dependant.

4.2.2 Recommender WCF Client

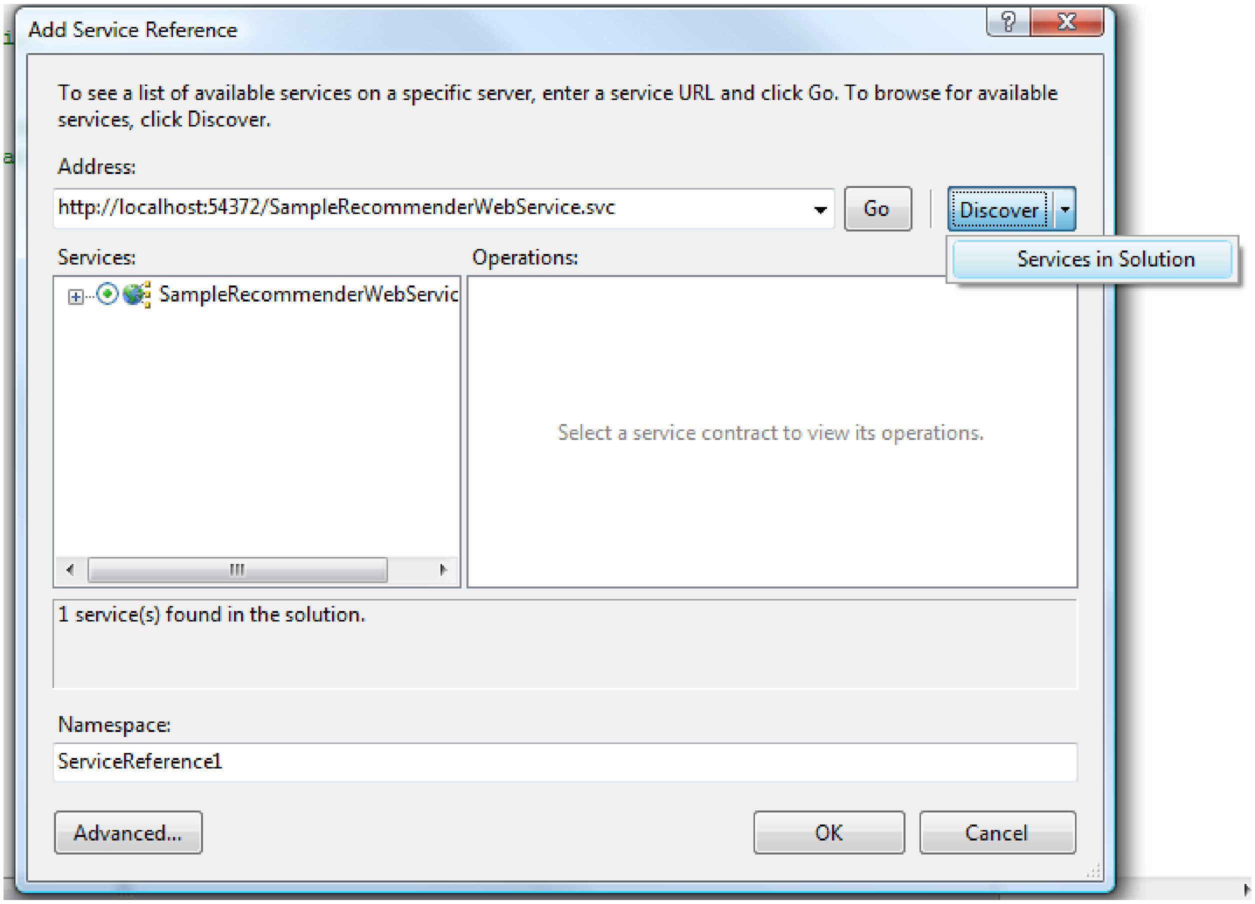
The WCF Recommender client must be run in association with the Recommender WCF Service. That is, the client will not work as expected unless the Service is available to fufil requests.

The Web Service class object is automatically generated by Visual Studio, using "Add Service Reference." This can be done as follows.

1. Right-click on the references component for the project



2. Next, configure the wizard to discover a service in your project



To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.

Address: http://localhost:54372/SampleRecommenderWebService.svc

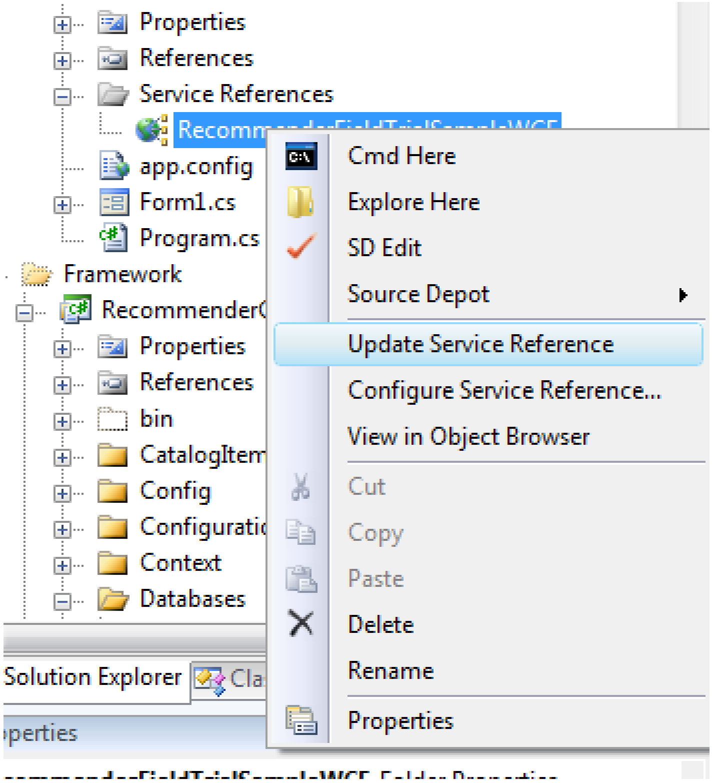
Services: SampleRecommenderWebServic

Operations: Select a service contract to view its operations.

1 service(s) found in the solution.

Namespace: ServiceReference1

3. Lastly, click “OK” to complete the process.
4. You have now gerated a WCF client for your WCF Service.
5. To consume this service
    a. Create a new instance of the SampleRecommenderWebServiceClient object
    b. Call the GetRatingsForUser method, with the appropriate parameters
6. To update this service, for example, after an interface change
    a. Right-click on the Service Reference in your project, and select update



4.2.3 Adding a New Method to the Sample

Adding a new interface to the WCF sample is quite simple. You only need to add a new method to your SVC file, and that will be automatically available as a web service.

1. Open ...\RecommenderFramework\RecommenderFieldTrialSampleWCF\SampleRecommenderWebService.svc.cs
    a. Add a new public method, with the “OperationContract” attribute

```
 [OperationContract]
public UserPrediction[] GetPredictionsForUser(int userId)
```

    b. Add any new business objects, attributing the class with the “DataContract” attribute, and the properties with the “DataMember” attribute

```
 [DataContract]
public class UserPrediction
{
    string title;
    double predictedRating;

    /// <summary>
    /// Gets or sets the predicted rating.
    /// </summary>
    /// <value>The predicted rating.</value>
    [DataMember]
    public double PredictedRating
```

```
            {
                get { return predictedRating; }
                set { predictedRating = value; }
            }

            /// <summary>
            /// Gets or sets the title.
            /// </summary>
            /// <value>The title.</value>
            [DataMember]
            public string Title
            {
                get { return title; }
                set { title = value; }
            }
        }
```

2. Update the WCF Client Service Reference, as described above (right click on the service reference and select update)
3. You will now see the new web service method available for use in your client application.

4.3 Deploy WCF Sample to IIS Web Server

4.3.1 Prerequisites

Prerequisites for the system

- IIS 5.x , 6.x or 7.x (free)
    - http://msdn.microsoft.com/en-us/library/ms181052(VS.80).aspx

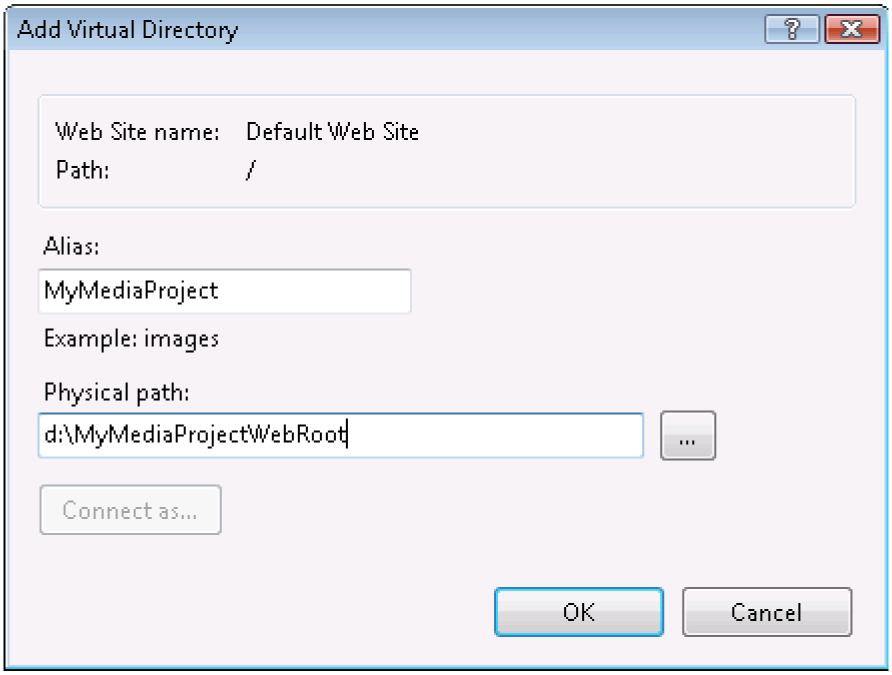
4.3.1.1 IIS Configuration

1. Enable IIS, if it is not already installed
    a. http://msdn.microsoft.com/en-us/library/ms181052(VS.80).aspx
2. Enable WCF extensions
    a. Run the following tool to enable WCF extensions in IIS
        i. %windir%\Microsoft.NET\Framework\v3.0\Windows Communication Foundation\ServiceModelReg.exe

4.3.1.2 IIS Default Web Site and Recommender Web Service Virtual Directory

1. Run the IIS configuration tool
    a. %SystemRoot%\system32\inetsrv\InetMgr.exe
2. Verify a default website has been enabled
    a. You should be able to access HTTP://<MachineName> and receive a default web page, such as this:

b. Add a new virtual directory for your Recommender sample code
    i. Right Click on “Default Web Site”
    ii. Select “Add Virtual Directory”
    iii. You will see a dialog like this

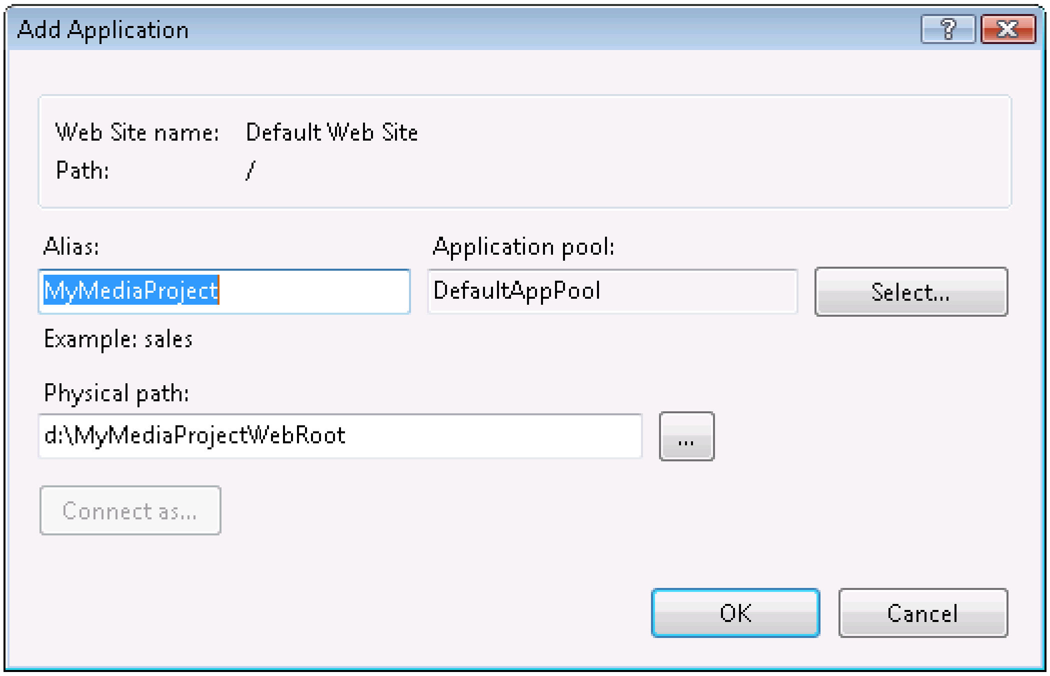


Web Site name: Default Web Site
Path: /

Alias: MyMediaProject
Example: images

Physical path: d:\MyMediaProjectWebRoot

        iv. Fill in the alias and physical path as appropriate
    c. Convert the virtual directory from above, to an Application
        i. Right click the new virtual directory, created previously, and select "Convert to Application," giving a dialog similar to this:



Web Site name: Default Web Site
Path: /

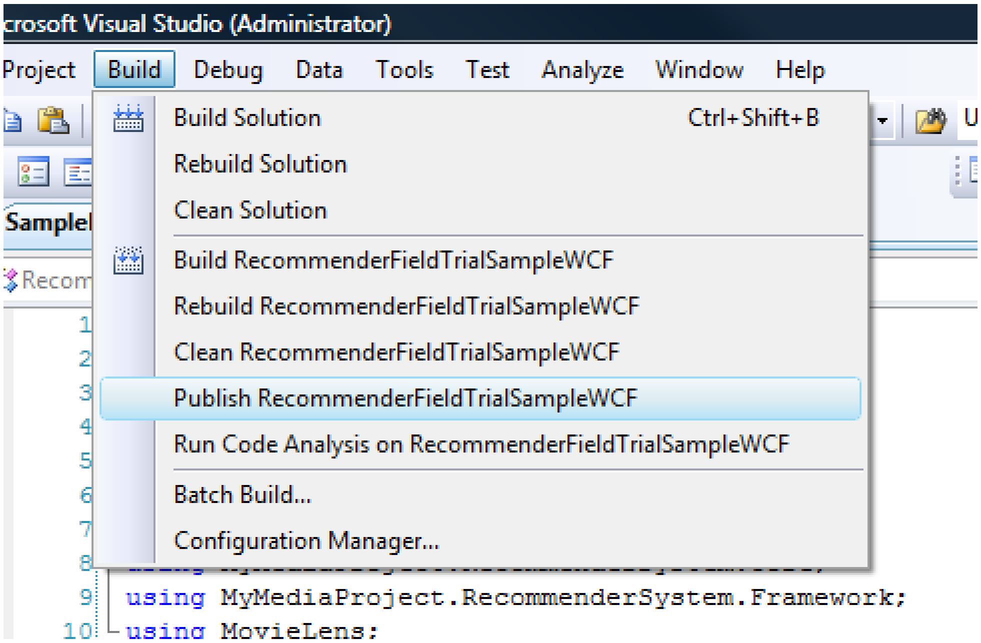
Alias: MyMediaProject
Application pool: DefaultAppPool
Example: sales

Physical path: d:\MyMediaProjectWebRoot

        ii. The defaults should be appropriate; verify that are consistent with your previous alias and physical path values

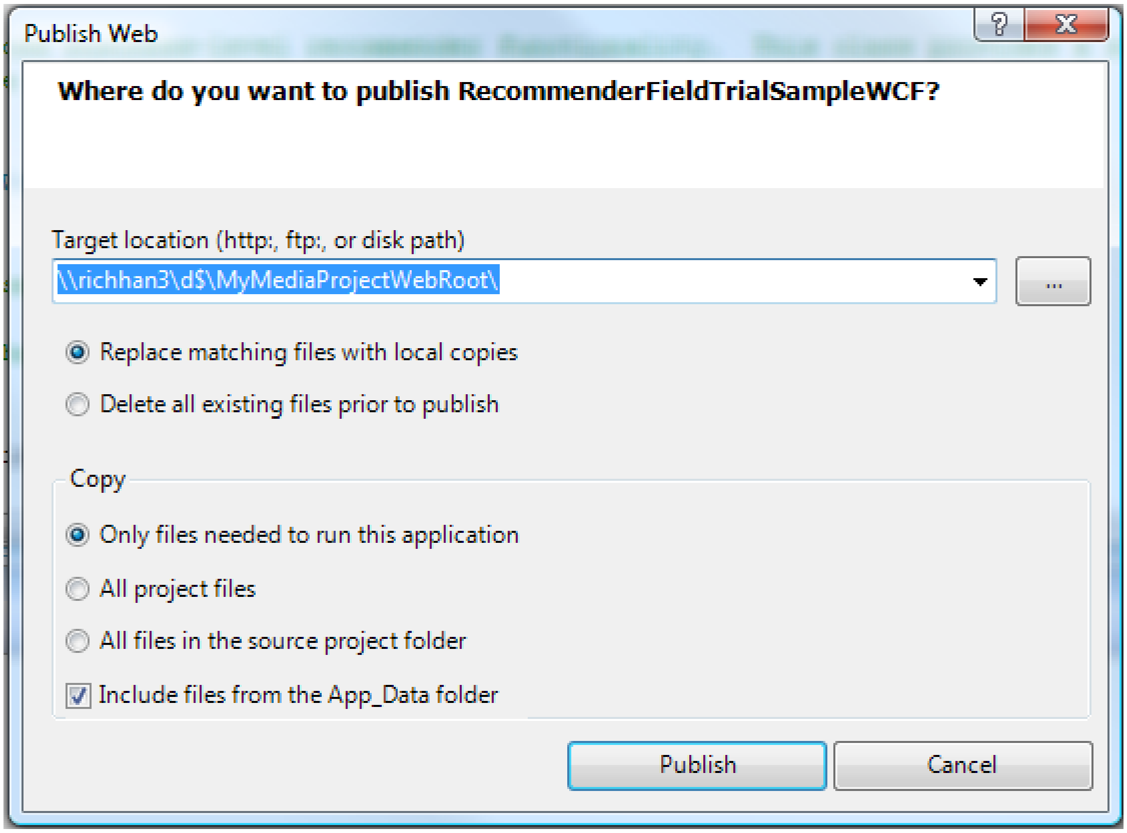
4.3.1.3 Deploy Recommender Web Service from Visual Studio

1. Using the "Physical Path" from step 2.b.iv, above, deploy the solution to the physical directory of your Recommender sample
    a. From Visual Studio, with the "SampleRecommenderWebService.svc.cs" file active, select "Build->Publish RecommenderFieldTrialSampleWCF," like this:


crosoft Visual Studio (Administrator)
Project
Build
Debug
Data
Tools
Test
Analyze
Window
Help
Build Solution Ctrl+Shift+B
Rebuild Solution
Clean Solution
Build RecommenderFieldTrialSampleWCF
Rebuild RecommenderFieldTrialSampleWCF
Clean RecommenderFieldTrialSampleWCF
Publish RecommenderFieldTrialSampleWCF
Run Code Analysis on RecommenderFieldTrialSampleWCF
Batch Build...
Configuration Manager...
Sample
Recom

```
using MyMediaProject.RecommenderSystem.Framework;
using MovieLens;
```

b. Which will give a dialog similar to this:



Where do you want to publish RecommenderFieldTrialSampleWCF?

Target location (http:, ftp:, or disk path)
\\richhan3\d$\MyMediaProjectWebRoot\

● Replace matching files with local copies
○ Delete all existing files prior to publish

Copy
● Only files needed to run this application
○ All project files
○ All files in the source project folder
☑ Include files from the App_Data folder

c. Replace “target location” in the dialog above, with the HTTP path, or physical path of your IIS virtual directory

   i. Note, you must have FrontPage Server Extensions to deploy directly to the web server. Deploying to the physical path should always work.

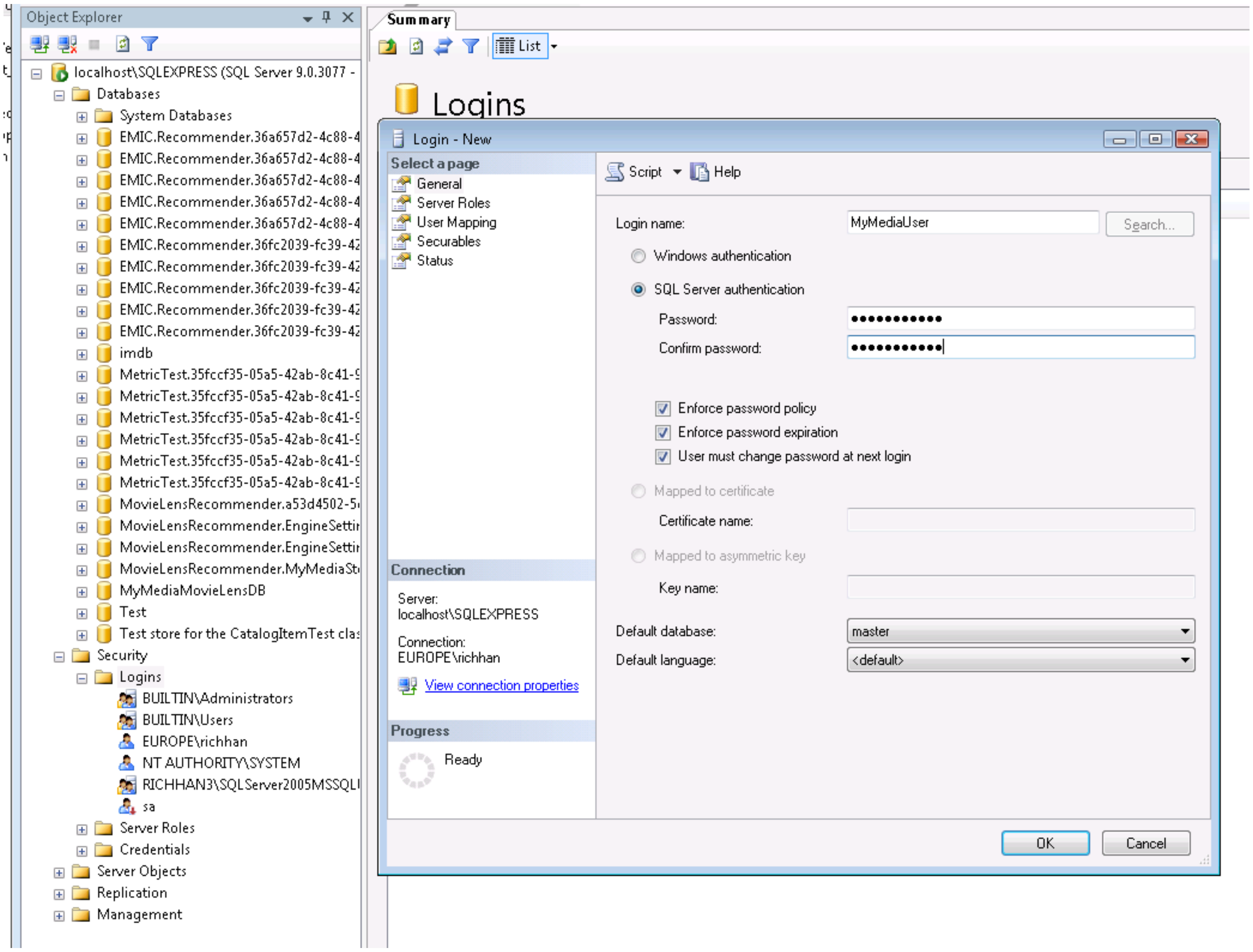
4.3.1.4 Database Connection, Configuration, and User Accounts

There are a number of ways to configure web services in the web server, to allow appropriate access to database resources.

Creating a Database User Account for MyMedia

One way of allowing access to database resources is to enable a user account which has read/write priviliages to the database.

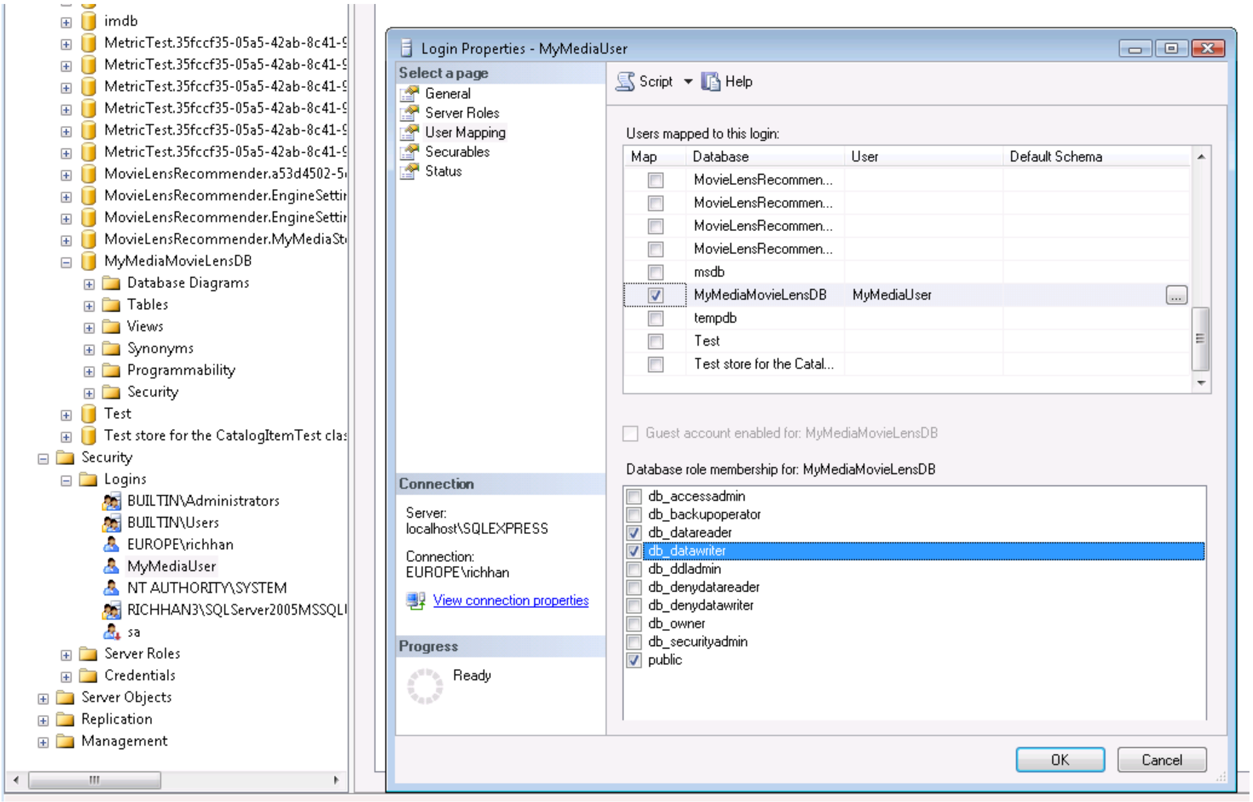
1. Open the Sql Server Management Studio application
2. Connect to your database server
3. Create a new user in your database server
    a. Under the Security->Logins panel, by right clicking on “Logins->New Login...” This will give you a diaglog like this:



Login name: MyMediaUser
○ Windows authentication
● SQL Server authentication
Password: ●●●●●●●●●●●●
Confirm password: ●●●●●●●●●●●●
☑ Enforce password policy
☑ Enforce password expiration
☑ User must change password at next login
○ Mapped to certificate
Certificate name:
○ Mapped to asymmetric key
Key name:
Default database: master
Default language: <default>
Server: localhost\SQLEXPRESS
Connection: EUROPE\richhan

   b. Select SQL Server Authentication
      i. NOTE: By default, Sql does not support Sql Server Authentication; you must configure this (see below for more information)

4. Assign the New User Account Priviliages to the Database
   a. Before the recently created user account can access the database, the permissions must be defined
      i. On the user account, right click and select properties
      ii. In the “User Mappings” section, select the appropriate database
      iii. Give the user the following permissions
         1. db_datareader
         2. db_datawriter
      iv. The resulting permissions should look like this:


imdb
MetricTest.35fccf35-05a5-42ab-8c41-
MetricTest.35fccf35-05a5-42ab-8c41-
MetricTest.35fccf35-05a5-42ab-8c41-
MetricTest.35fccf35-05a5-42ab-8c41-
MetricTest.35fccf35-05a5-42ab-8c41-
MetricTest.35fccf35-05a5-42ab-8c41-
MovieLensRecommender.a53d4502-5
MovieLensRecommender.EngineSetti
MovieLensRecommender.EngineSetti
MovieLensRecommender.MyMediaSt
MyMediaMovieLensDB
Database Diagrams
Tables
Views
Synonyms
Programmability
Security
Test
Test store for the CatalogItemTest cla
Security
Logins
BUILTIN\Administrators
BUILTIN\Users
EUROPE\richhan
MyMediaUser
NT AUTHORITY\SYSTEM
RICHHAN3\SQLServer2005MSSQL
sa
Server Roles
Credentials
Server Objects
Replication
Management
Login Properties - MyMediaUser
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Connection
Progress
Ready
View connection properties
OK
Cancel

Users mapped to this login:

Map	Database	User	Default Schema
☐	MovieLensRecommen...		
☐	MovieLensRecommen...		
☐	MovieLensRecommen...		
☐	MovieLensRecommen...		
☐	msdb		
☑	MyMediaMovieLensDB	MyMediaUser	
☐	tempdb		
☐	Test		
☐	Test store for the Catal...		

☐ Guest account enabled for: MyMediaMovieLensDB

Database role membership for: MyMediaMovieLensDB

- ☐ db_accessadmin
- ☐ db_backupoperator
- ☑ db_datareader
- ☑ db_datawriter
- ☐ db_ddladmin
- ☐ db_denydatareader
- ☐ db_denydatawriter
- ☐ db_owner
- ☐ db_securityadmin
- ☑ public

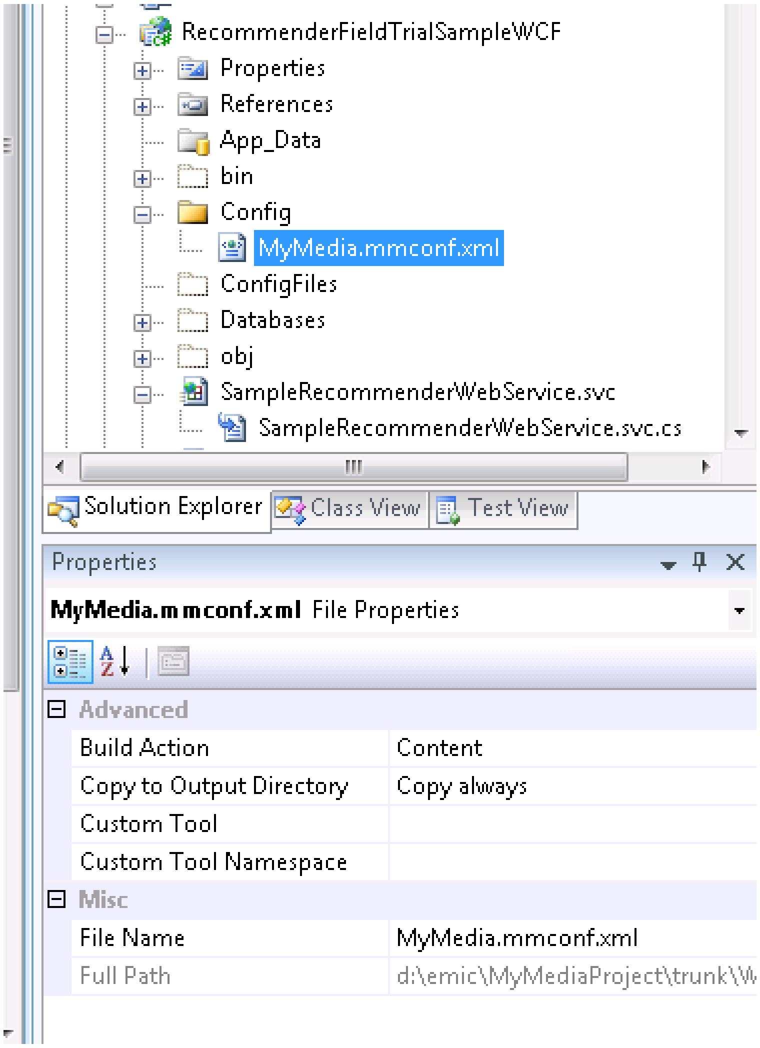
Server:
localhost\SQLEXPRESS

Connection:
EUROPE\richhan

5. Enable Sql Server Authentication
   a. Right click on your database connection, and select “Properties”
   b. Under the “Security” node, in the “Server Authentication” section, enable “Sql Server and Windows Authentication Mode”
   c. Restart the database server for the changes to take effect

Configure the MyMedia Framework Database Connection

The database configuration string can be specified in the MyMedia configuration file. This should be added in a project folder called “Config”, starting with the defaults specified in the framework MyMedia.mmconf.xml file. In the project settings, this should be copied all of the time. Like this:



MyMedia.mmconf.xml File Properties

Advanced	
Build Action	Content
Copy to Output Directory	Copy always
Custom Tool	
Custom Tool Namespace	
Misc	
File Name	MyMedia.mmconf.xml
Full Path	d:\emic\MyMediaProject\trunk\\

In this file, you can specify the database connection string.

1. Open the MyMedia.mmconf.xml file
2. In the properties list, modify the xml property, “DBConnectionString”
    a. Set the value to the appropriate database connection string
    b. This should include the user account and password created previously
    c. This connection string should look something like:

<Property Name="DBConnectionString" Value="**Data Source=localhost\SQLEXPRESS;Integrated Security=True;Initial Catalog=MyMediaMovieLensDB;** ***UID=MyMediaUser; PWD=mymediauser;***" Description="The DB connection string together with the name of the database"

```
Default="Data Source=localhost\SQLEXPRESS;Integrated Security=True;Initial
Catalog=MyMediaDB" />
```