HIS document 4

Configuring web services for an observations database (version 1.0)

A setup guide

January 2008

Prepared by:

David Valentine and Tom Whitenack
San Diego Supercomputer Center
University of California, San Diego

Tim Whiteaker and Ernest To
Center for Research in Water Resources
University of Texas at Austin

Distribution

Funding

Funding for HIS Server development was provided by the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) under NSF Grant No. EAR-0413265. In addition, much input and feedback has been received from the CUAHSI Hydrologic Information System development team. Their contribution is acknowledged here.

Technical Support

For questions and support regarding HIS Server, please contact:

Tom Whitenack
University of California, San Diego
San Diego Supercomputer Center, MC 0505
9500 Gilman Drive
La Jolla, CA 92093-0505
twhitenack@sdsc.edu

Table of Contents

1.0 Introduction

The CUAHSI Observations Data Model (ODM) is a standard schema for storing hydrologic observations time series data. An Observations Database (OD) is created when scientists load their data into the ODM schema. CUAHSI provides two ways of publishing data in an OD: through web services for programmatic access, and through a map interface for access via a web browser. This document consists of an exercise which shows how to set up a web service for your data.

Definition:

- ODM – Observations Data Model – a standard schema for storing hydrologic observations time series data
- OD – Observations Database – a database which follows the ODM schema

To reduce effort in setting up web services for your data, CUAHSI has developed a generic ODM web service, which may be configured to work with your observations data. In other words, if you choose to use the ODM schema for storing your observations data, nearly all the work of building a web service on top of that schema has already been done for you!

For the exercise, the user installs a generic ODM web service, and configures it to read data from a sample OD.

The procedure here involves a bit of configuration and setting permissions for file and folder access. A rough outline of the steps involved is shown below.

A. Install an OD.
B. Set permissions for accessing the database.
C. Set up web service folder.
    a. Create the folder.
    b. Copy the generic ODM web service to the new folder.
    c. Set folder permissions.
    d. Set up web sharing.
D. Allow anonymous access via IIS.
E. Configure and test the web service.

2.0 Requirements

Software:

- Micosoft Windows 2003 server or Windows XP
- IIS with ASP.Net
- .Net 2.0 Framework (Redistributable Package)
- SqlServer Express, or SQL Server Standard
- Installation zip file containing Generic OD web service support files from http://water.sdsc.edu/softwarereleases/GenericODWebServices/.

Data:

- A sample Observations Database populated with data from Little Bear River from the OD web site: http://water.usu.edu/cuahsi/odm/files/LittleBearRiver.zip.

User Privileges:

- User with administrator privileges

3.0 Installing an OD

The generic OD web service publishes data from an Observations Database, typically on the same machine as the web service itself. This database must be configured with the appropriate permission so that a web service can access it.

You could start with your own OD if you already have one populated with data. However, for the sake of being able to follow along with this document, we recommend you use a sample OD that has already been populated with data.

The database that you will use contains sensor data from the Little Bear River in northern Utah.

1. Download a sample OD from http://water.usu.edu/cuahsi/odm/files/LittleBearRiver.zip.
2. Extract both files in the zip file to wherever you store your SQL databases. If you are using an HIS Server machine, a typical location to use is the Data disk partition (O: drive), in the SQLData folder, where you would make a new folder for your data. For example, you could place the files in O:\SQLData\LittleBearRiver\.
3. Navigate to http://www.cuahsi.org/his/odm.html. There, you will find a link to a document called Instructions for Attaching Blank SQL Server Schema. Use those instructions as a guide to attach the LittleBearRiver database.

4.0 Setting Database Permissions

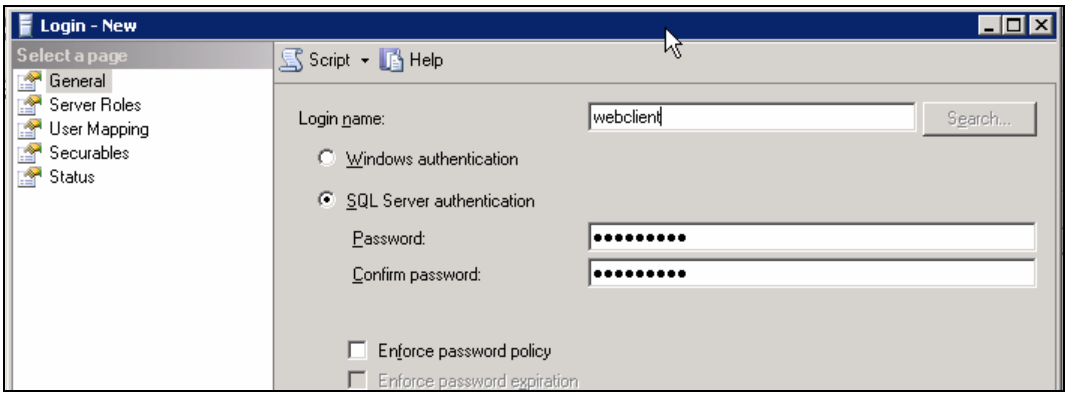
In order for a web service to be able to return data from your database, you must tell SQL Server to allow that database to be accessed by the web service. With SQL Management Studio still open from your work in the previous section, you will now set database permissions for a special login called **webclient**. This webclient login provides a layer of security for your application, and is safer than giving a standard login such as network_service access to your database. The webclient login will only have read access to the databases used by the web services.

1. If you haven't done so already, log into SQL Management Studio.

If the webclient login isn't already available in SQL Server, you will need to create it.

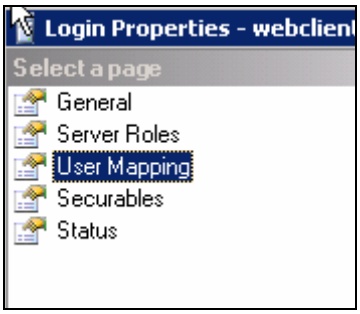
2. In the Object Explorer, expand Security, and select Logins. If you do not see webclient in the list of logins, right click on Logins and click New Login. Otherwise, skip to step 5.

3. In the New Login dialog, select SQL Server authentication. Then uncheck Enforce password policy if you don't want to manage this password expiring, etc.
4. Set the login name as "webclient", and choose a password. For the purposes of these screenshots, the password I will use is webclient. Click OK to close the dialog.

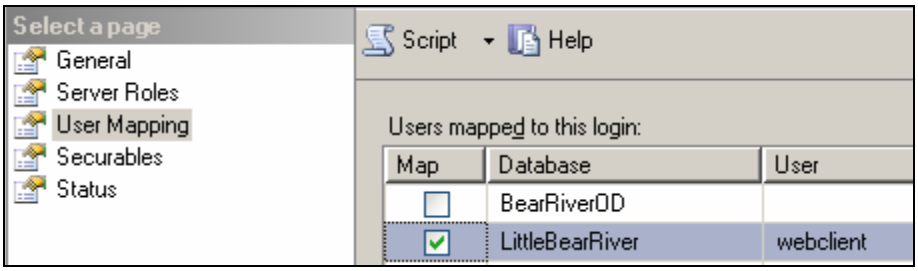


The webclient login should now appear in the list of logins.

5. Right click on the webclient login and click Properties.
6. Select User Mapping on the left side of the dialog. This displays the databases this login can access.

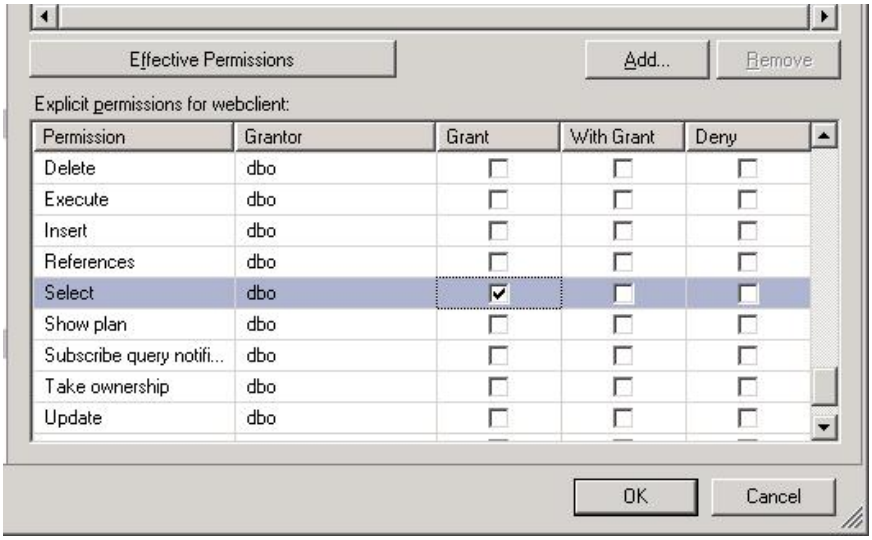


7. Put a check in the box next to your database (LittleBearRiver), and then click OK to close the Login properties page.



8. In the Object Explorer, expand the Databases folder. Then right click on your database (LittleBearRiver), and click Properties.
9. In the Database Properties dialog, select the Permissions page on the left.

10. On the right, where the current users are displayed, select the webclient user.
11. In the list of effective permissions at the bottom of the dialog, scroll down and place a check in the box next to Connect, and then next to Select, in order to Grant permissions to webclient. The check should go in the column labeled "Grant".



12. Click OK to close the Database Properties dialog.
13. Close SQL Server Management Studio.

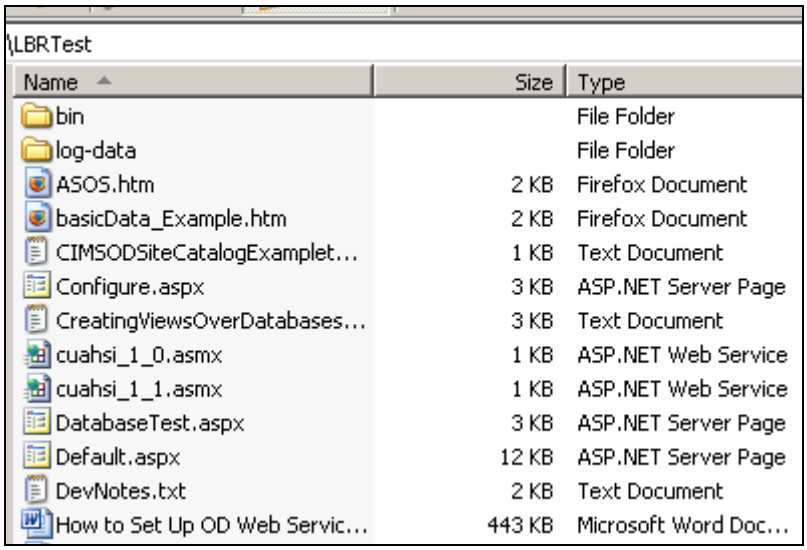
5.0 Setting Up the Web Service Folder

With database permissions set, now you will set up the folder that will contain the web service which accesses the database.

If you are using a preconfigured HIS Server, recall that HIS Server comes configured with disk partitions dedicated to different HIS components. While the ODM database finds a happy home on the Data disk partition (O:\), the best place to put a new web service is within the HIS Software partition (P:\). If you are using a server that was not preconfigured as an HIS Server, you can install the web service in a location of your choosing.

A generic web service for accessing an OD has been created for CUAHSI, and can be downloaded, installed, and configured for your OD.

1. In Windows Explorer, create a folder in which you will place the web service. For this exercise, you can call the folder LBRTest. (LBR stands for Little Bear River).
2. Navigate to http://water.sdsc.edu/softwarereleases/GenericODWebServices/.
3. Download GenericODWS-1-0_Release.zip.
4. Unzip all files into the LBRTest folder.



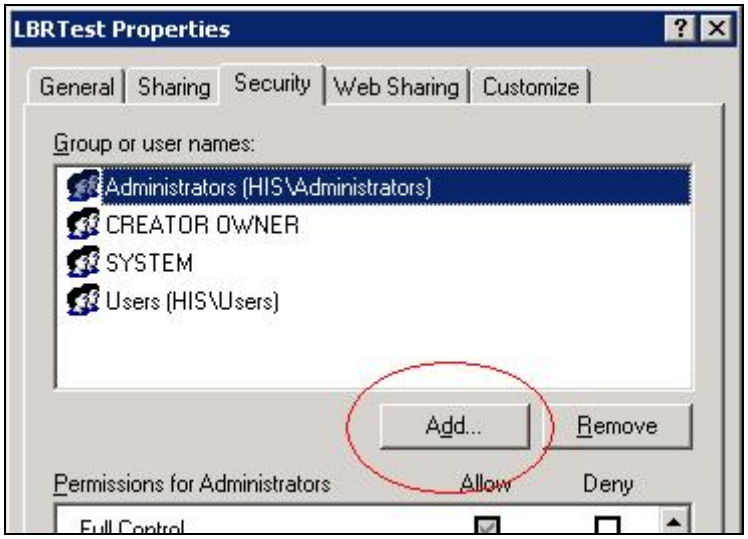
Now you will set permissions on the folder, and set the folder up so that the web service can be accessed from the Internet.

5. Right click on the LBRTest folder and click Properties.
6. Click the Security tab.

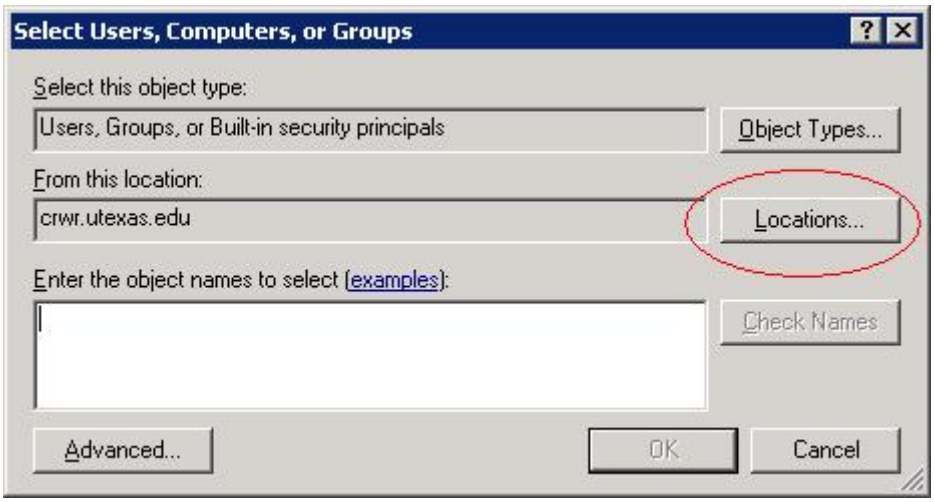
You will now add “Everyone”, “ASPNET”, and “IIS_WPG” to the list of users for this folder.

Note: If you are using Windows XP, you will not add IIS_WPG. Instead, you will add IWAM_ComputerName, where ComputerName is the name of your computer. So if your computer name is VM-XP, the user you will add is IWAM_VM-XP.

7. Click Add.

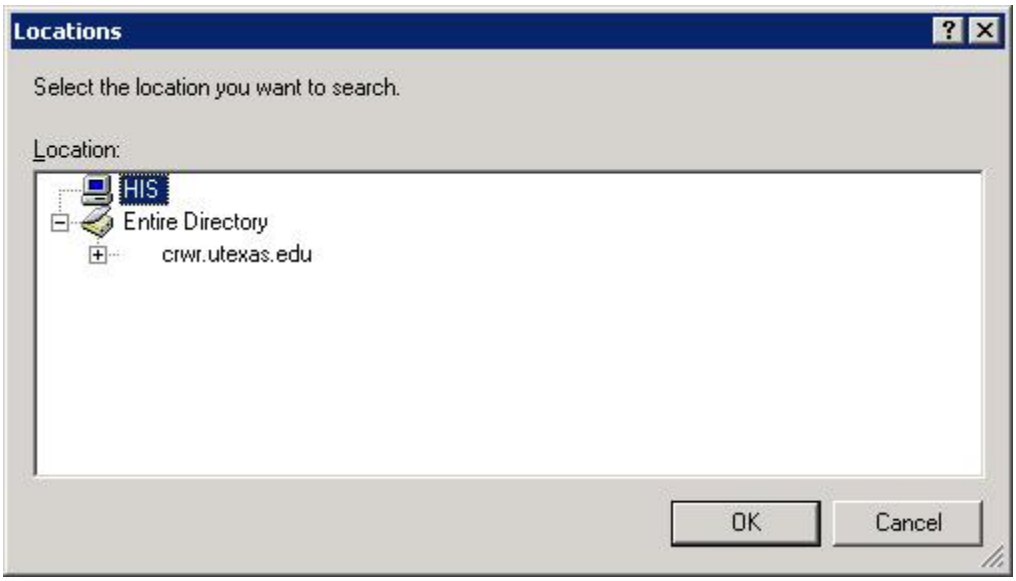


8. In the Select Users, Computers, or Groups dialog, click Locations. This lets you define the location where the accounts of interest are defined.

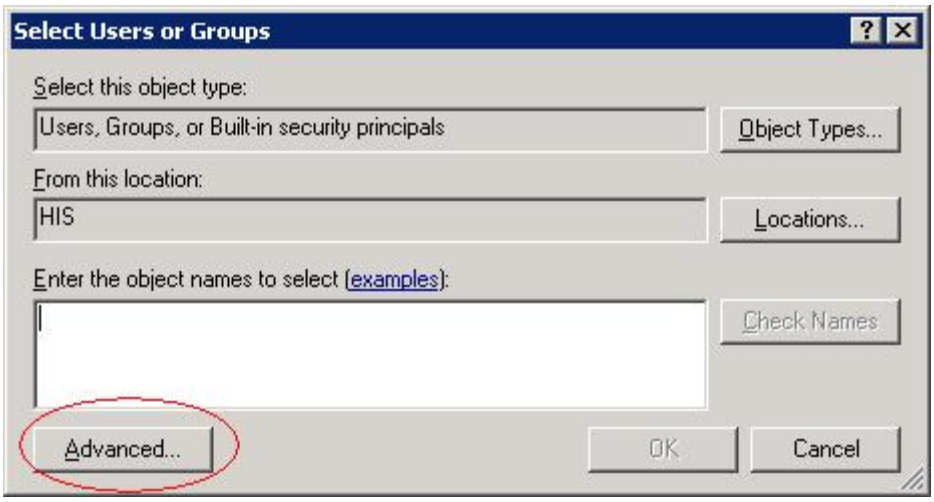


In my case, I was then prompted for the username and password to connect to the default location, which is our intranet here at UT. I don't need the intranet, as the desired accounts are defined on the HIS Server itself, so I clicked cancel to cancel out of the log in dialog. Then I was presented with a Locations dialog.

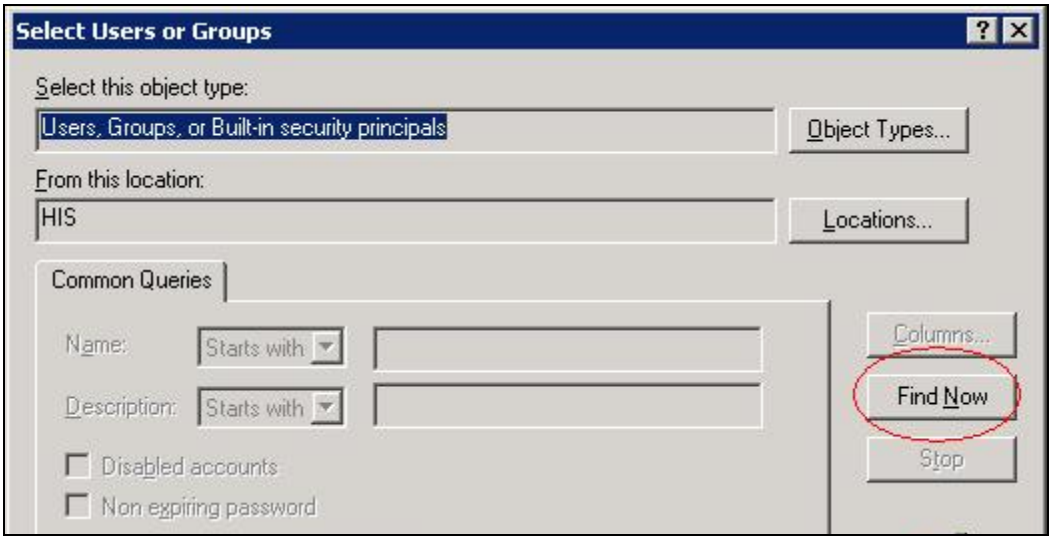
9. Select the name of your computer, and click OK.



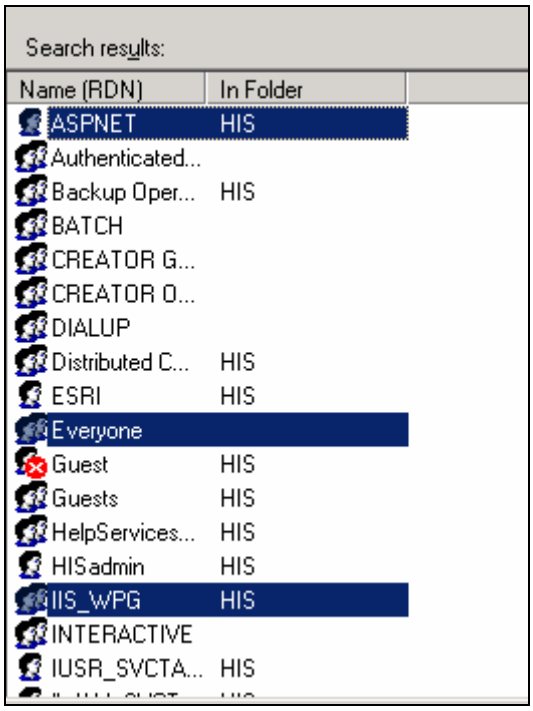
10. Now that the location is set to your computer, click the Advanced button. This lets you browse for the user name of interest.



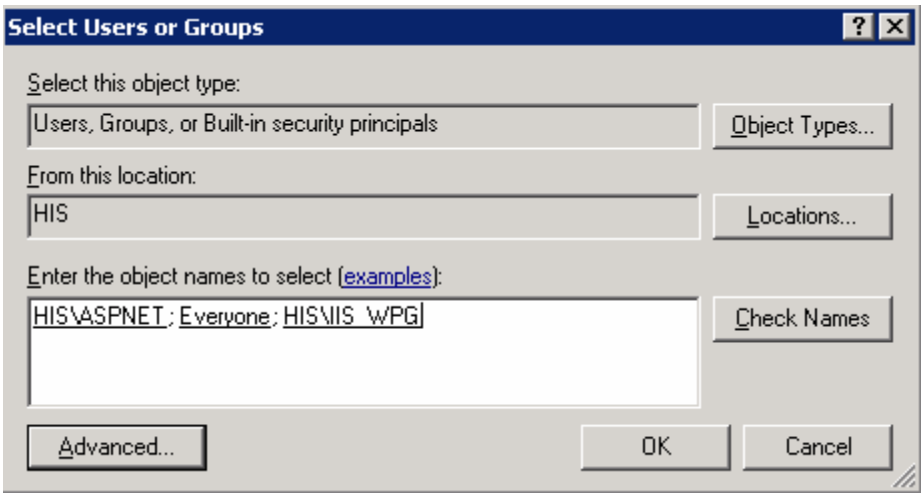
11. In the dialog that opens, click Find Now. This populates the dialog with a list of user names.



12. Select Everyone, ASPNET, and IIS_WPG (or IWAM_ComputerName for XP users) from the list, and then click OK. Tip: Use the Ctrl key as you click on users to select more than one at the same time.

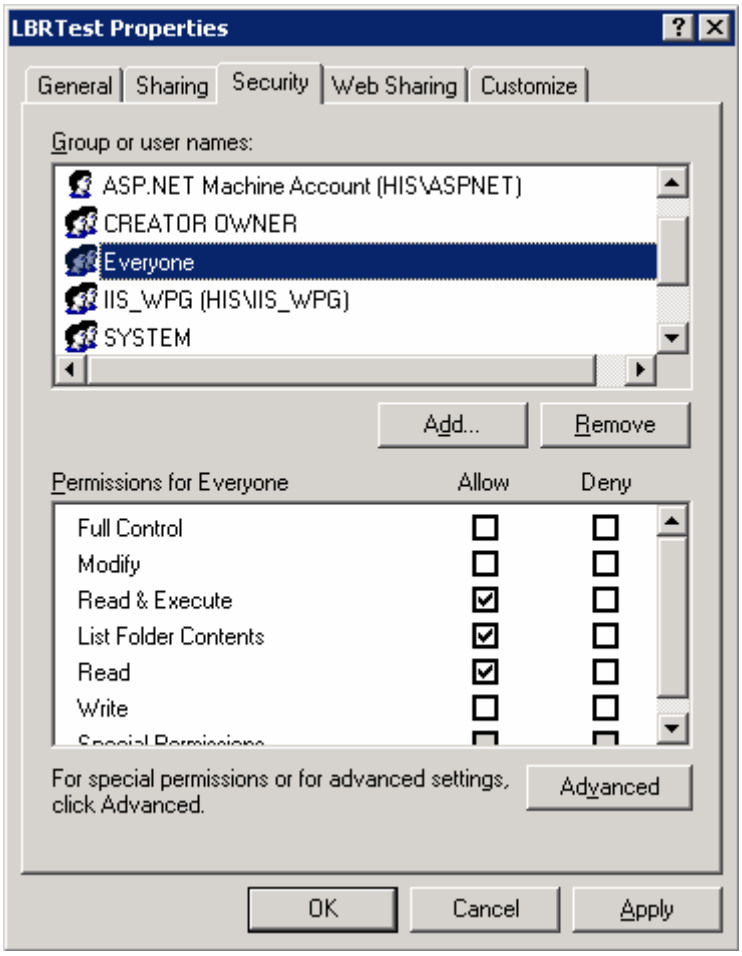


13. The accounts now appear in the Select Users or Groups dialog. Click OK.



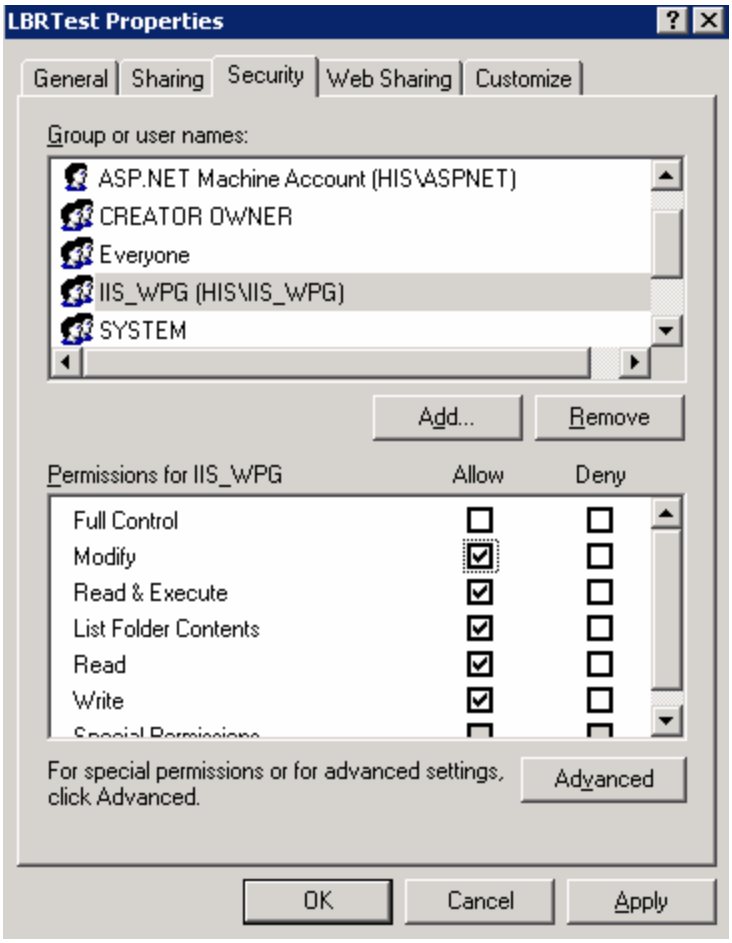
Now that the users are added, you will set their permissions. "Everyone" needs "Read & Execute" access, "IIS_WPG" needs "Modify" access, while "ASPNET" needs "Full Control".

14. In the LBRTest Properties dialog, select the user named Everyone and make sure this user has Read & Execute access.

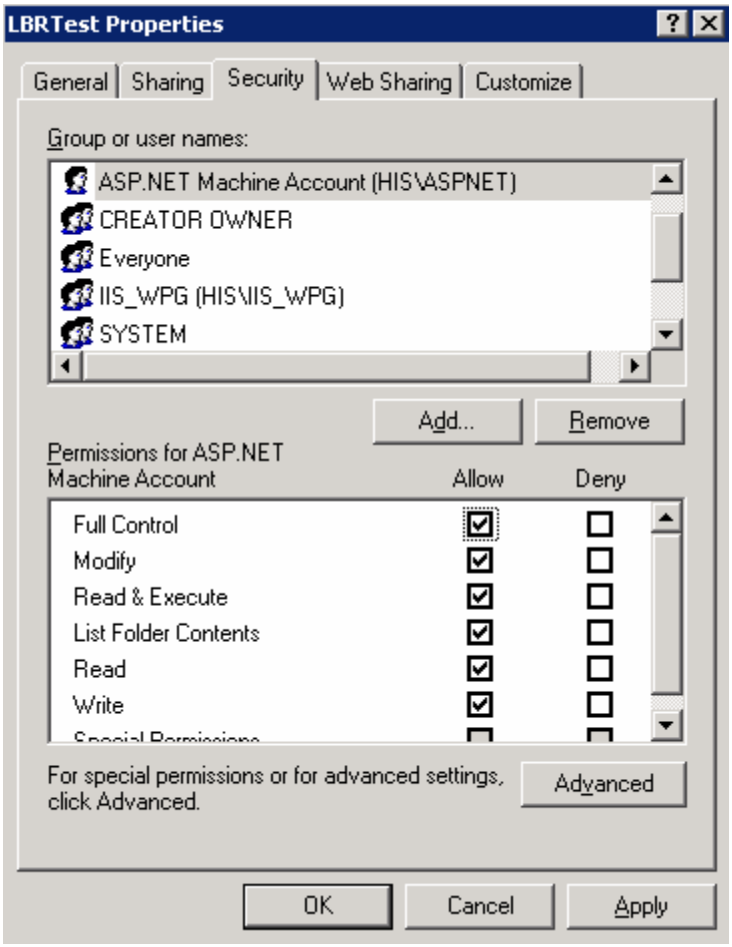


15. Select the IIS_WPG user (or IWAM_ComputerName for XP users) and give this user Modify access by placing a check next to Modify.

Note that when you click Modify, some other check boxes will be filled in automatically. This is expected. Don't worry about it. Basically, check boxes get filled in if they are lower in the hierarchy of permissions than the item that you click on.

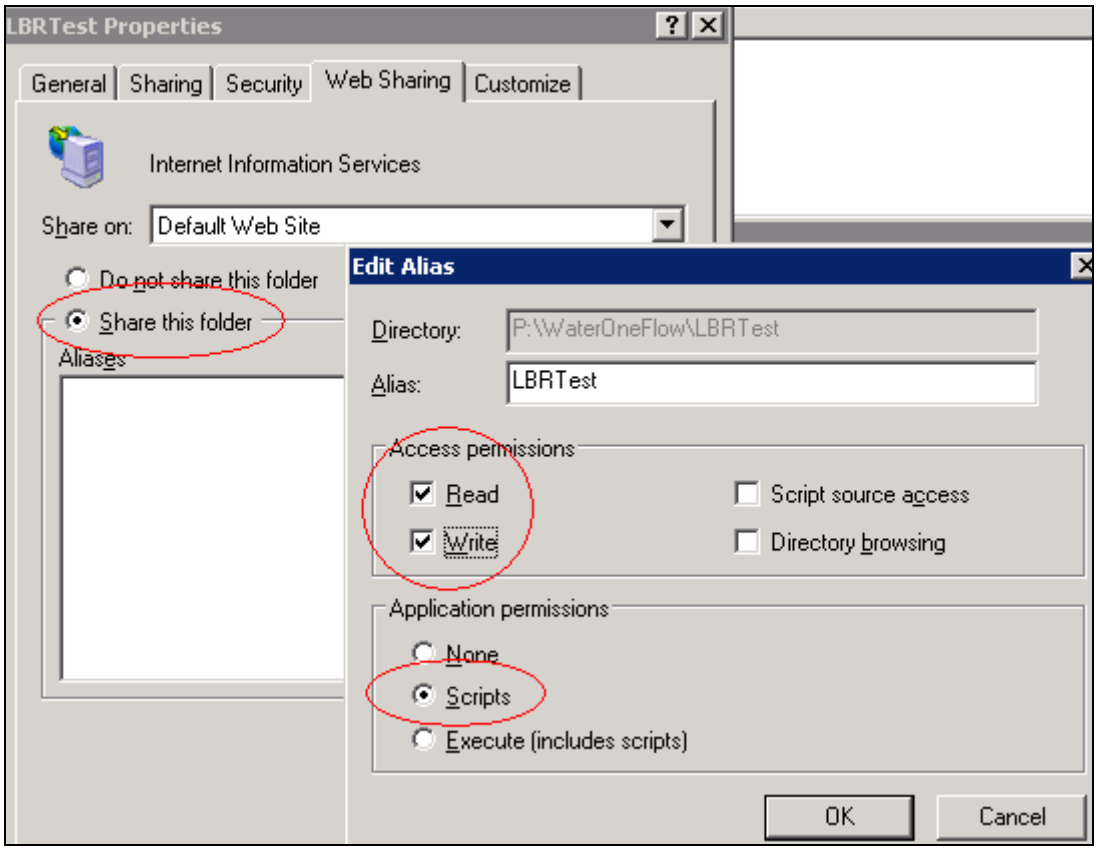


16. Select the ASPNET user and give this user full control by placing a check next to Full Control.



With permissions set, now you will set up the folder to be shared on the web.

17. Click the Web Sharing tab.
18. Select the “Share this folder” option. When you do so, the Edit Alias dialog will open.
19. In the Edit Alias dialog, make sure both “Read” and “Write” are checked, and that scripts are enabled. When you place a check next to “Write”, you may be prompted with a message box warning you about granting write access. Just click OK to dismiss the message.

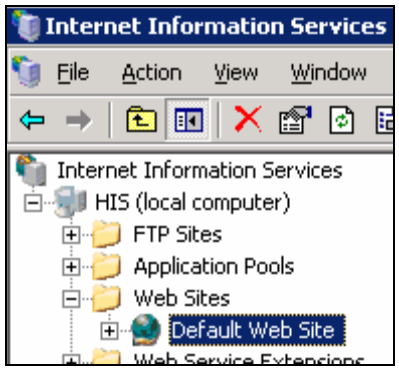


20. Click OK to close the Edit Alias dialog.
21. Click OK to close the properties dialog.

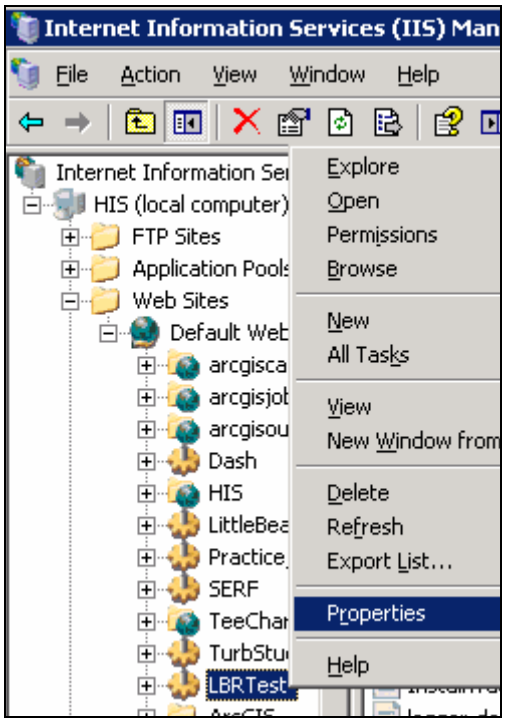
6.0 Allowing Anonymous Access

The way that your computer is exposed to the Internet is managed by Internet Information Services (IIS). You'll use IIS to allow your ODM web service to be accessed by anonymous users over the web. (New to IIS? Don't worry…this part is quick!)

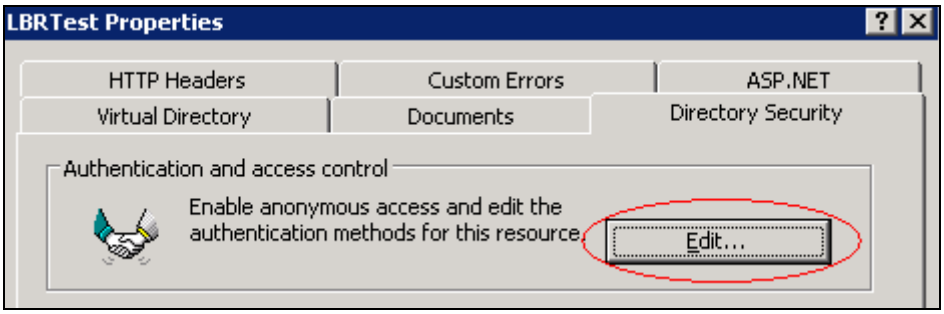
1. Start IIS. (Start --- Administrative Tools --- IIS Manager, or Start --- Control Panel --- Administrative Tools --- IIS Manager)
2. Expand the IIS tree for your computer to see Web Sites. Expand Web Sites to see Default Web Site. This is the default web site that is set up on your computer when IIS is installed, hence the name Default Web Site.



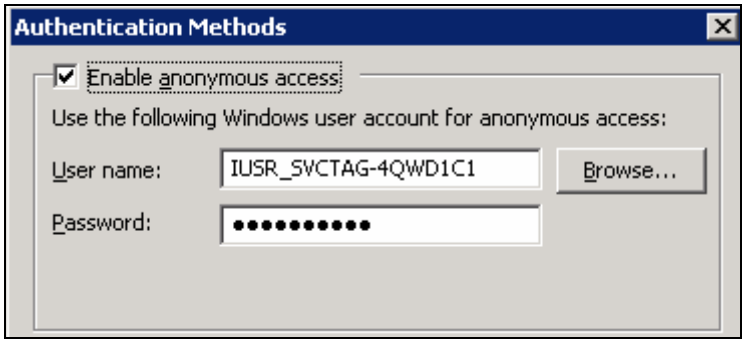
3. Expand Default Web Site, and you will see the web applications on your computer. Notice that there is an LBRTest folder. This folder was added here when you chose to share the LBRTest folder in the previous section.
4. In IIS, right click the LBRTest folder and click Properties.



5. Click the Directory Security tab.
6. In the Authentication and access control frame, click Edit.

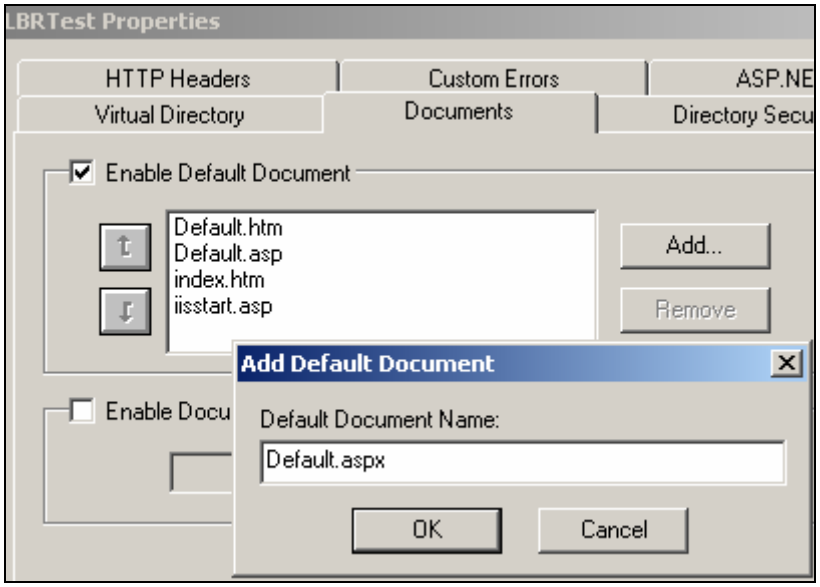


7. Place a check next to Enable anonymous access, and click OK.



At this point, you should also check to make sure that by default, this "web site" that you've just created will open with a Default.aspx file by default. Default.aspx is the web service file. If you don't set this, you will get a Directory Listing Denied error when you try to view the web service page in a browser.

8. Click the Documents tab.
9. If you don't see Default.aspx as one of the documents, add it by clicking the Add button, typing in Default.aspx, and clicking OK.



10. Click OK to close the LBRTest Properties dialog.
11. Close IIS.

7.0 Configuring and Testing the Web Service

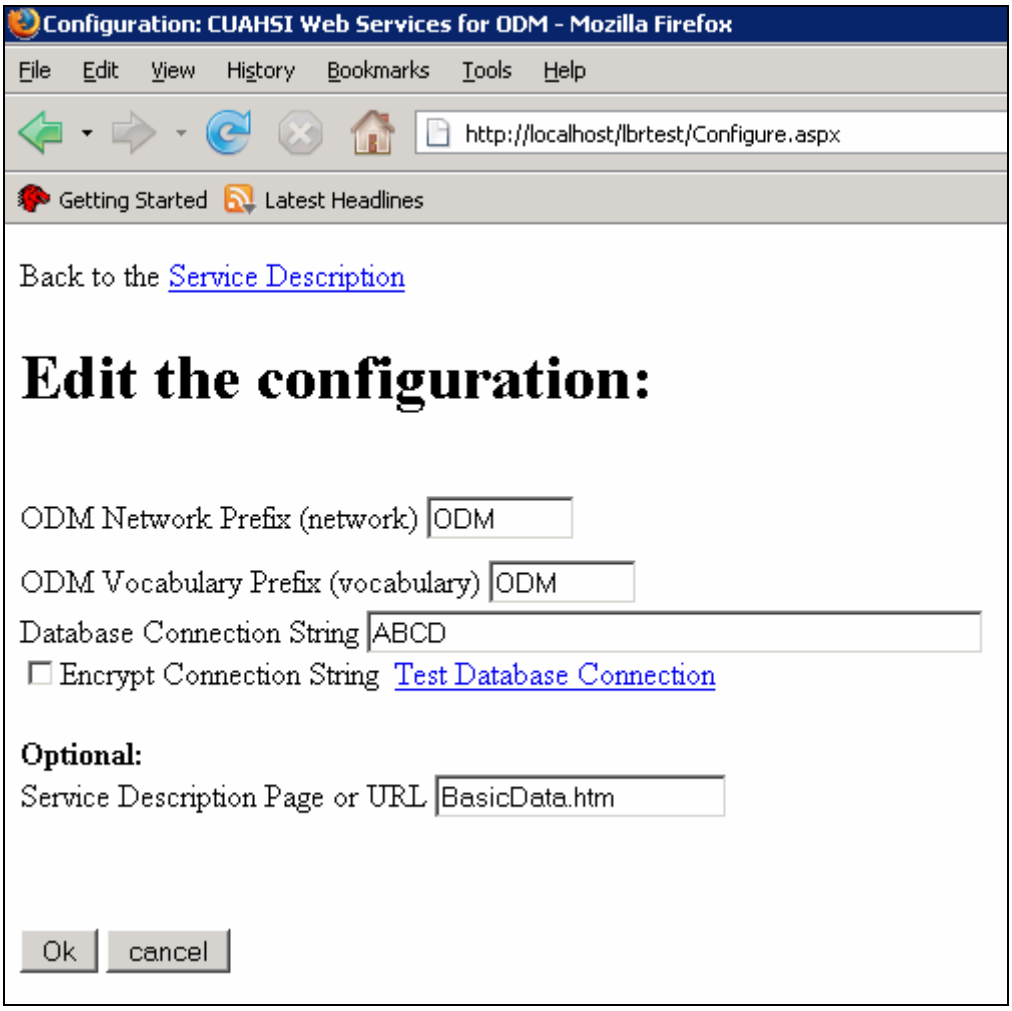
The web service should now be exposed on the Internet. To finish up, you will configure the service and run a quick test.

1. Start a web browser.
2. Navigate to your web service site on your computer. This address begins with "http://localhost/", followed by the name of the folder as seen in IIS. For this exercise, that address is http://localhost/LBRTest/.

A page for the web service opens.

3. Click the Configuration link.

A page opens that lets you configure some properties of the web service.



Note: You can only change configurations with this web page when you are logged into the machine locally. For security reasons, users accessing this web site from other machines won't be able to see this configuration page.

Now let's talk about some of these settings.

Network

The Network prefix is a short identifier for the observation network that your sensors or sites belong to. For example, if all of your sensors are for a water quality monitoring project for the Columbia River watershed, you might name your network "ColumbiaWQ".

You might be wondering how the Network setting comes into play. This setting affects how you ask for site information when calling the ODM web service from a program. Specifically, the web service methods affected are GetSites/GetSitesObject, GetSiteInfo/GetSiteInfoObject, and GetValues/GetValuesObject.

The syntax the any WaterOneFlow service must follow regarding site information, is "Network:SiteCode", where Network refers to the Network prefix, and SiteCode is a unique identifier for the site in the network. For example, if I'm a program and I'm asking a WaterOneFlow web service for information about site 08158000 in the USGS NWIS network(which happens to be the Colorado River at Austin), I would ask for "NWIS:08158000".

The reason why the Network prefix is required, is because a given web service may be providing access to more than one observation network. For example, NWIS and EPA sites may be stored in the same database, or at least provided access to via the same web service. The Network prefix allows us to distinguish between site code 08158000 in the NWIS network, and site code 08158000 in the EPA network, which may represent completely different sites and locations.

For our example with the Little Bear River data, we'll just use LBRTest as the Network prefix.

Vocabulary

The Vocabulary prefix is a short identifier for the vocabulary used to describe the variables in your observations network. For example, if I have a variable called Texonium with a unique code of TXN, I'd like to know if TXN is a term that the EPA uses, or perhaps the USGS uses, or perhaps I just made it up and it's my own vocabulary. The Vocabulary doesn't have to be the same as the Network. For example, suppose I like the way NWIS describes variables, and I've used the same variable codes in my own network. My Network prefix might be "MyNetwork", but my Vocabulary prefix could be "NWIS".

The syntax for using Vocabulary when calling a WaterOneFlow web service is "Vocabulary:VariableCode", where VariableCode is the unique identifier for the variable of interest, and Vocabulary is the system to which that given VariableCode belongs. The web service methods affected by Vocabulary are GetVariableInfo/GetVariableInfoObject, and GetValues/GetValuesObject.

For our example with the Little Bear River data, we'll just use LBRTest as the Vocabulary prefix.

Note: Once you've decided on a Network and Vocabulary, you might want to make note of those so that you can keep track of them.

Database Connection String

This is the trickiest part of configuration. You need to provide the web services with a database connection string for your ODM database. This string basically has everything that the web service needs to know in order to connect to your database.

There is a grand discussion of connection strings in the documentation at http://river.sdsc.edu/wiki/SQL%20Server%20Connection%20Strings.ashx.

For **Windows Server 2003** with **SQL Server 2005**, your connection string will look like the following:

```
Data Source = ComputerName; Initial Catalog = LittleBearRiver; User
ID=webclient; Password = webclient; Persist Security Info = True
```

You would substitute the name of your computer (e.g., VM-XP) for ComputerName above. The above connection string says to use the local computer as the data source, and to access the LittleBearRiver database from within that data source. The connection is established with the credentials of the webclient user.

For **Windows Server 2003** with **SQL Express**, your connection string will look like the following:

```
Data Source = .\sqlexpress; Initial Catalog = LittleBearRiver; User
ID=webclient; Password = webclient; Persist Security Info = True
```

You would have noticed if you were using SQL Express, because the term SQLEXPRESS would have appeared in on the login page when connecting to SQL Management Studio earlier.

For Windows XP, your connection string will look like the following:

```
Data Source = localhost; Initial Catalog = LittleBearRiver; User
ID=webclient; Password = webclient; Persist Security Info = True
```

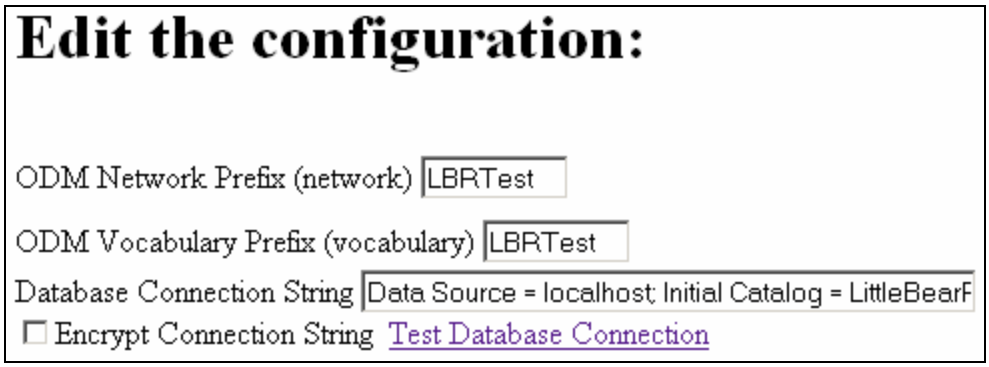
Once you've got the connection string down, you might want to make note of it somewhere.

If you want, you can encrypt the connection string with the check box provided on the configuration web page. You would want to do this if you were providing any important passwords in your database connection string.

Service Description Page

If you want, you can provide a web page with more information about your observations network. For this exercise, we will not worry about this extra web page.

4. Fill in the configuration properties as below:
    a. Network – LBRTest
    b. Vocabulary – LBRTest
    c. Database Connection String – [Use the example above, or whatever string works for your particular setup]
    d. (Ignore the other configuration options for now)



5. **Important:** Click **OK** first to commit the configuration. Then click **Test Database Connection**. If the connection worked, you should see a new web page listing some sites and variables as read from your database. If the page does not show up properly, then check your database connection string and try again.

This should display up to 10 sites

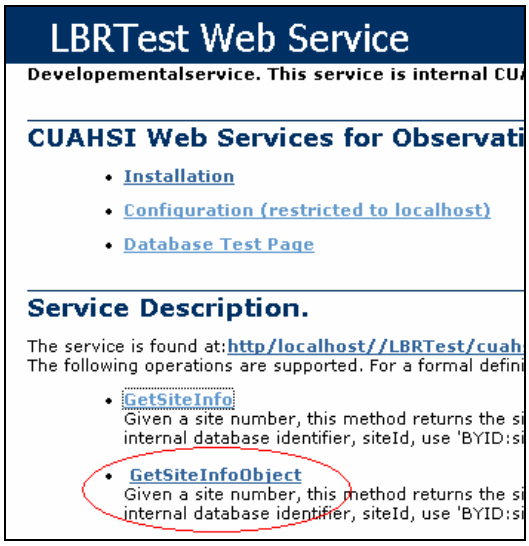
SiteCode	SiteName
USU-LBR-Mendon	Little Bear River at Mendon Road near Mendon, Utah
USU-LBR-Paradise	Little Bear River at McMurdy Hollow near Paradise, Utah
USU-LBR-ExpFarm	Utah State University Experimental Farm near Wellsville, Utah
USU-LBR-SFLower	South Fork Little Bear River below Davenport Creek near Avon, Utah
USU-LBR-EFLower	East Fork Little Bear River at Paradise Canal Diversion near Avon, Utah
USU-LBR-EFWeather	Little Bear River Upper Weather Station near Avon, Utah

This should display up to 10 variables

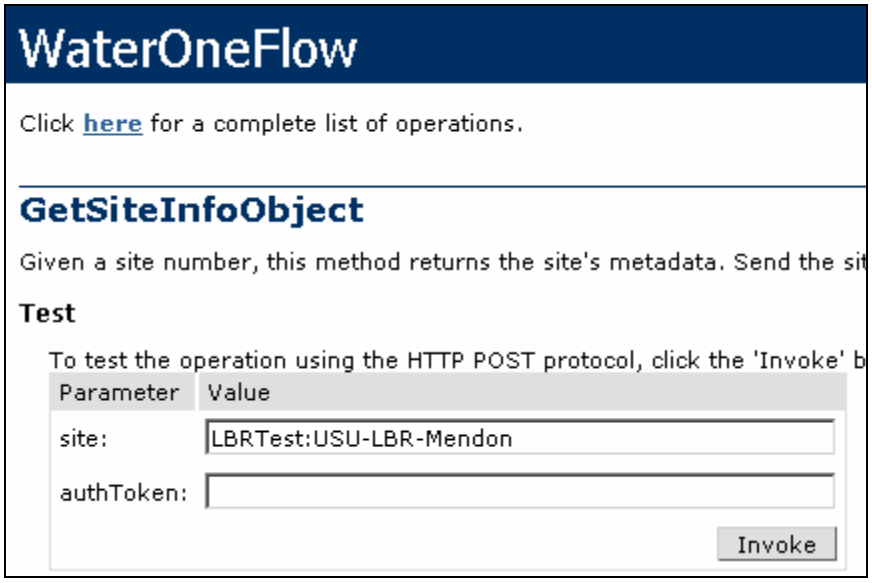
VariableCode	variableName
USU3	Battery voltage
USU4	Turbidity

6. Record one of the site codes listed on the test page. You'll use that site code to perform one more quick test on the web service.
7. Go back to http://localhost/LBRTest/.

After the header on this page, you see a list of the web service methods that are available. You'll call GetSiteInfoObject to test the service.



8. Click GetSiteInfoObject. This brings you to a page where, if you are operating from localhost, you can actually input some arguments to test the web method.
9. Enter the site code you recorded earlier. Don't forget the Network prefix, LBRTest.



10. Click Invoke.

If the service worked, you should see an XML page with information about the site. This XML format is a WaterML response from the web service, where WaterML is a standard language developed by CUAHSI for communicating water information over the web. The XML may be hard for a human to read, but it is actually quite easy to write a program to read it. Since WaterOneFlow web services are designed to be accessed by programs, WaterML is a good format for returning data from our web service.

Congratulations! Your web service is up and running.

8.0 Advanced Settings

The sections below discuss advanced configuration options. These options provide more control over the operation and display of the web service, but are not required just to get the service up and running.

- Your system administrator may ask that the connection string be encrypted so that users cannot casually examine the web.config file.
- You can log queries and errors for your services.
- You can edit the application configuration, manually.
- You can provide basic information about your web service.

8.1 Enable Encrypted Connection String Settings (Optional)

You can enable encryption of the connection string in the application web.config file. Enabling this will secure the connection parameters from prying eyes. From localhost go to the page: http://localhost/{Your Application}/configure.aspx.

8.2 Configure logging (Optional)

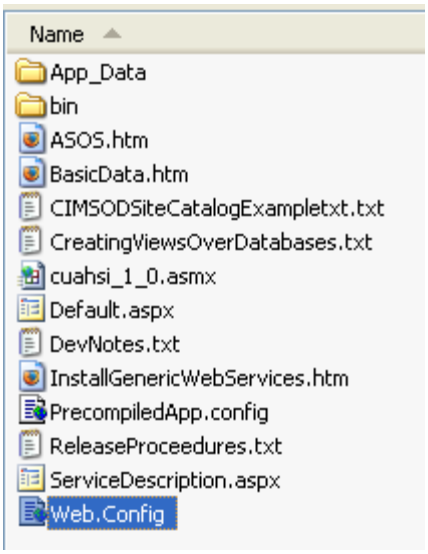
Logging uses a codebase called log4net (http://logging.apache.org/log4net/). There are two logs created, 1) a query log (log-data\query-log.txt), and 2) webapp-log.txt. The query-log.txt contains details on the queries, and webapp-log.txt can contain detailed debug information. Server logs are configured to send data to SDSC and write local error information to a directory called log-data. If this directory is not writable or present, then no local error information will be available. To change, open "wateroneflow.logging.log4net" and edit the following:

```
<appender name="QueryLogFileAppender"
type="log4net.Appender.RollingFileAppender">
    <file value="log-data\query-log.txt" />
<!-.. ->
</appender>
```

8.3 Manual Editing of Web.Config

Apart from using the Configuration Web Page to edit the web service configuration (i.e. network name, vocabulary and connection string), one may manually edit the configuration file (i.e. Web.Config). The follow steps illustrate how you may do this.

1. In the folder for your web service, locate the Web.Config file, and open it either with a text editor or Visual Studio 2005.



web config 1

2. In Web.Config, set the network and vocabulary for your service by editing the appropriate <add> elements. The network refers to the observation network to which your database pertains, such as “NWIS Daily Streamflow”. The vocabulary refers to the framework within which the terms that you use apply, such as “NWIS”. You can leave these as the default value of “ODM” if you wish. The “GetValuesName” can be used by applications as a hint of what to call your network in a menu. It should be a brief name (<25 characters).

```
<add key="network" value="ODM"/>
<add key="vocabulary" value="ODM"/>
<add key="GetValuesName" value="Local OD GetValues" />
```

3. Edit the connection string for your database. To find the connection string, locate the <connectionStrings> element. Then find the <add> element that is a child of <connectionStrings>, which has a name of “ODDB”. This element has an attribute called “connectionString”, where you will provide the connection string for your database. This is the same connection string discussed in the section of this document regarding web service configuration via the configuration web page.

4. Save and close Web.Config

8.4 Create BasicData.htm and Edit

A file is included which is appended to the web service description page when that page is viewed with an Internet browser. By default, that page is called "BasicData.htm" and gives information about your service, such as example parameters that can be used for testing the service. Edit this page to describe your service and data.

1. Copy BasicData_Example.htm to BasicData.htm.

2. Edit BasicData.htm. You may put whatever content you like in this file.
3. Save and close BasicData.htm.

As an example, BasicData.htm has been edited to show a snippet of what is available in the LittleBearRiver database.

Information for LBRTest web service

This web service provides access to the Little Bear River database of surface water monitoring data.

Some example sites and variables available in this database include:

Site	Variable	BeginDate	EndDate
LBRTest:USU-LBR-Mendon	LBRTest:Turbidity	2005-08-04	2007-09-26
LBRTest:USU-LBR-Paradise	LBRTest:Temperature	2005-06-29	2007-09-26

Notes:
This is the Demo Database.