CUAHSI

universities allied for water research

ODM Web Data Loader

An application for loading data into the CUAHSI Hydrologic Information System Observations Data Model

March 2012

Prepared by:

Jeffery S. Horsburgh
Utah Water Research Laboratory
Utah State University

Distribution

The ODM Web Data Loader application and all associated source code and documentation are available at the following URL: http://his.cuahsi.org/.

The ODM Web Data Loader software, source code, and documentation are provide free of charge under the Berkeley Software Distribution (BSD) license. Please see the following license information:

Disclaimers

Although much effort has been expended in the development and testing of the ODM Web Data Loader application, errors and inadequacies may still occur. Users must make the final evaluation as to the usefulness of the ODM Web Data Loader for their application.

The ODM Web Data Loader application and this software manual are based upon work supported by the National Science Foundation under Grants No. 03-26064, and 06-10075. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Acknowledgements

The team of engineers and software developers that developed the ODM Web Data Loader application includes:

Jeffery S. Horsburgh, Research Assistant Professor, Civil and Environmental Engineering, Utah Water Research Laboratory, Utah State University, Logan, UT.

David G. Tarboton, Professor, Civil and Environmental Engineering, Utah Water Research Laboratory, Utah State University, Logan, UT.

Kim Schreuders, System Administrator, Utah Water Research Laboratory, Utah State University, Logan, UT

Stephanie Reeder, Programmer, Utah Water Research Laboratory, Utah State University, Logan, UT

Technical Support

There is no formal ongoing support for this freely distributed open source software. However, we are interested in feedback. If you find errors, have suggestions, or are interested in any later versions, please contact:

Jeffery S. Horsburgh
Utah State University
8200 Old Main Hill
Logan, UT 84322-8200
jeff.horsburgh@usu.edu

Table of Contents

1.0 Introduction and Software Description

The CUAHSI Hydrologic Information System (HIS) Project has developed information technology infrastructure to support hydrologic science. One of the components of the HIS is a point Observations Data Model (ODM) (Horsburgh et al., 2008; Tarboton et al., 2008), which is a relational database schema that was designed for storing time series data. The purpose of the ODM is to provide a framework for optimizing data storage and retrieval for integrated analysis of information collected by multiple investigators. The CUAHSI HIS ODM has been implemented by a number of data publishers and researchers throughout the country, and these researchers are using ODM as a mechanism for publication of individual investigator data and for registering these data with the National HIS.

The ODM Web Data Loader (ODMWDL) application was created to allow administrators of local instances of the ODM and other users to load data into an ODM database using a web application interface. The development of the ODMWDL application has several advantages. First, ODMWDL protects the security and consistency of an ODM database because it provides users with a set of tools for validating and loading their data into ODM. This minimizes the potential for human caused errors in loading these data into an ODM database. The ODMWDL input file formats are similar to the table structures in ODM, but they do provide users with some flexibility in specifying the required metadata. Users do not need to perform any specialized programming to parse and load the data, and the ODMWDL ensures that the data are fully qualified with valid metadata when they are loaded.

In some cases, collaborating researchers from multiple institutions have had difficulty in managing the loading of data into ODM databases due to firewalls between the database server hosting the ODM database and the people who wish to load the data. The ODMWDL provides a mechanism for loading data into an ODM database that is not blocked by most common firewalls – e.g., the ODMWDL runs within an organization's firewall using common ports and communication mechanisms that are not blocked by most firewalls and is available to any user with a web browser and an Internet connection.

1.1 General Functionality

The main objective of ODMWDL application is to provide data managers with a set of tools for loading data into an ODM database. The ODMWDL is implemented as a web application that is accessed using a web browser. ODMWDL is a file based data loader. It is capable of loading all of the individual tables within ODM separately from separate input files (i.e., one file per table), or it can do bulk data loading of all of the ODM tables from a single input file. In general, a single execution of ODMWDL loads a single file. To protect the integrity of data within an ODM database, the ODMWDL treats each input file as a single data loading transaction. Because of this, the ODMWDL will not load any data unless it can successfully load all of the data contained within the file.

1.2 Supported Data File Formats

ODMWDL was designed to load data from individual files. It currently supports comma delimited text files (.csv) that have a one row header that uses ODM field names in the header, followed by the data in subsequent rows. Each file to be loaded must conform to the input file templates provided in Appendix A of this document. These input templates generally conform to the table structure of ODM, with some flexibility for specifying alternative information for database generated IDs. For example, when loading data values, the SiteCode and VariableCode can be used rather than the SiteID and VariableID.

1.2 Platform and Minimum System Requirements

ODMWDL was designed to run on Microsoft Windows 2003 Server or 2008 Server based computers. It is recommended that machines running the ODMWDL software have at least 100 MB of free disk space and at least 1 gigabyte of RAM. In addition, computers running ODMWDL application must have the Microsoft .Net Framework Version 3.5 installed prior to installing the ODMWDL. Instructions for obtaining the .Net Framework Version 3.5 from Microsoft are included in the Installation Instructions section below.

ODMWDL is a web client application. It must be connected to an instance of the CUAHSI HIS ODM that has been implemented in Microsoft SQL Server 2005 or 2008. ODMWDL is capable of connecting to any ODM database to which it can make a direct SQL connection. The ODM SQL Server database can be located on the same machine as the ODMWDL application, or ODMWDL has the capability to connect to a remote ODM database provided that the database server name and ODM database name are known and the user has been given write access to, and SQL Server authentication information for, that server and database.

1.3 ODM Compatibility

The ODMWDL is compatible with ODM Versions 1.1 and 1.1.1. ODM and the ODM Design specifications documents can be obtained from the following URL: http://his.cuahsi.org.

2.0 Installation Information

2.1 Installation Prerequisites

Prior to running the ODMWDL installation, you must first install the Microsoft .Net Framework Version 3.5 (if it is not installed already). If you have Microsoft SQL Server 2008 installed, Version 3.5 of the .Net framework will be installed already. The .Net Framework Version 3.5 is free, and is required to run software applications developed in Microsoft's Visual Studio .Net 2008. Instructions for downloading and installing the .Net Framework Version 3.5 can be obtained from the Microsoft website via the following URL:

http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=22

Once the .Net Framework Version 3.5 has been installed, you can continue with the ODMWDL installation.

NOTE: ODMWDL requires that you have an ODM database implemented in Microsoft SQL Server 2005 or 2008. If you do not already have an instance of Microsoft SQL Server running, you can download and install Microsoft SQL Server 2008 Express from Microsoft for free. It is recommended that you download and install either the "Runtime with Management Tools" or the "Runtime with Advanced Services" version of SQL Server 2008 express. You can get these products and instructions for installing them at the following Microsoft URL:

http://www.microsoft.com/express/sql/download/

For instructions on how to attach a blank ODM database to an instance of Microsoft SQL Server, please consult the ODM documentation available at http://his.cuahsi.org.

2.1.1 Setting Up a Domain for Your HydroServer

Prior to installing the HydroServer Web applications – e.g., the HydroServer Website, the HydroServer Map Application, the Time Series Analyst, the WaterOneFlow Web Services, the HydroServer Capabilities Web Services, and the ODMWDL – you will want to create a domain for your HydroServer. You will need to work with the Information Technology professionals within your organization to help you create a domain. Once a domain has been created for your HydroServer, you can proceed in the setup of the HydroServer software.

In implementing the HydroServer Web applications, it is helpful to understand the structure of the overall deployment so that each of the pieces can be implemented correctly. The HydroServer Website was designed to be a parent, or root level, application within a domain that you set up for your HydroServer. For example, if you were to create a domain name for your HydroServer called "icewater.usu.edu," the URL for your HydroServer Website would be at the root level of that domain (i.e., http://icewater.usu.edu/ would be the URL for your HydroServer Website). Each of the other HydroServer Web applications was designed to be a child application of the HydroServer domain. The following shows how the other HydroServer Web applications would be implemented under the HydroServer Website within the same domain:

http://icewater.usu.edu/tsa/ - The Time Series Analyst
http://icewater.usu.edu/map/ - The HydroServer Map application
http://icewater.usu.edu/HydroServerCapabilities/ - The HydroServer Capabilities Web service

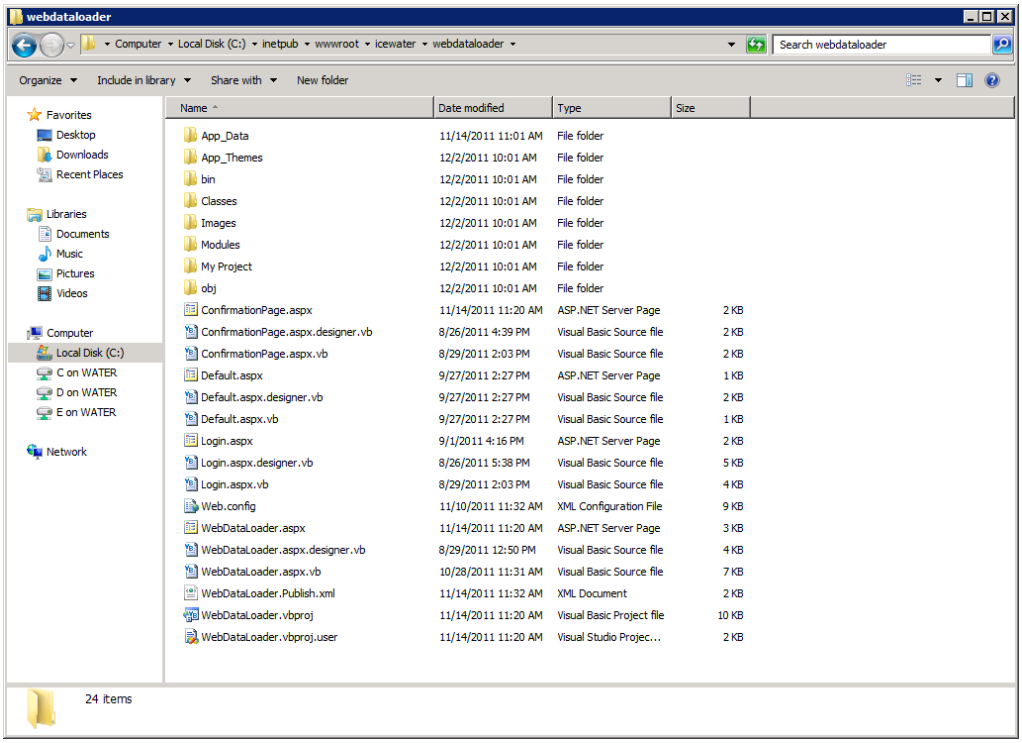
http://icewater.usu.edu/webdataloader/ - The ODM web data loader
http://icewater.usu.edu/LittleBearRiver/ - A WaterOneFlow web service for the Little Bear River experimental watershed
http://icewater.usu.edu/MudLake/ - A WaterOneFlow web service for data collected within Mud Lake at the Bear Lake National Wildlife Refuge

The above example assumes that your HydroServer is serving as the Web server running Microsoft Internet Information Services (IIS), as the database server running Microsoft SQL Server, and as the GIS server running ArcGIS Server. This doesn't have to be the case, though, and there is quite a lot of flexibility for the components of your HydroServer to be spread across multiple machines and implemented within multiple domains. In general, the HydroServer documentation assumes that you are assembling your HydroServer within a single domain.

2.2 Installing and Configuring the ODM Web Data Loader

Use the following steps to install and configure the ODMWDL. The following steps were written for a web server running Windows Server 2008 and IIS Version 7.

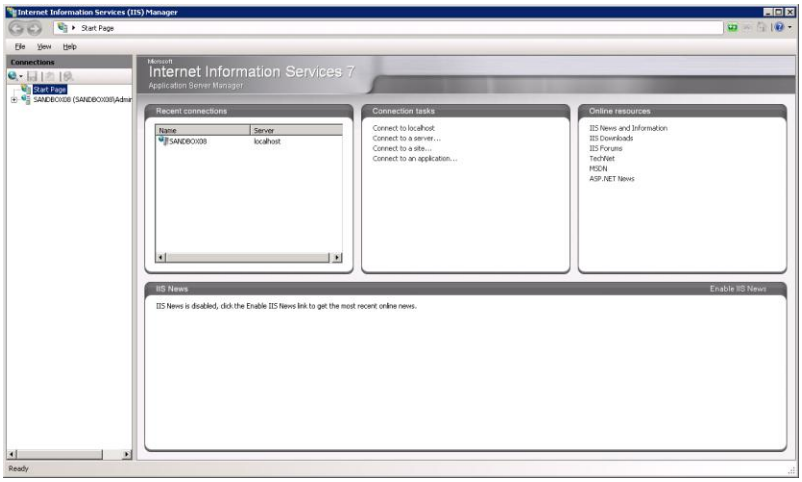
1. Copy the folder from the zip file containing the ODMWDL program files ("ODMWebDataLoader.zip") to a web application directory on your HydroServer. For this example, I have set up a folder called "icewater" under my "c:\inetpub\wwwroot\" path. I copied the folder containing the ODMWDL files from the zip file to the "icewater" folder and renamed it "webdataloader" (see the following figure).



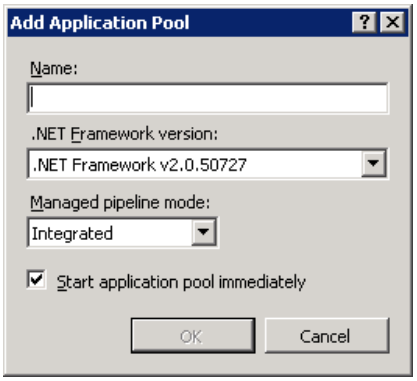
NOTE: In this example, I have put the ODMWDL files within a folder called "icewater" within my "c:\inetpub\wwwroot\" folder. I might do this if I were building a HydroServer within a domain called "icewater," e.g., if the URL for my HydroServer was something like http://icewater.usu.edu. This folder can also

contain the other Web applications that are part of HydroServer, including the HydroServer Capabilities Web service, the HydroServer Website, and the WaterOneFlow web services.

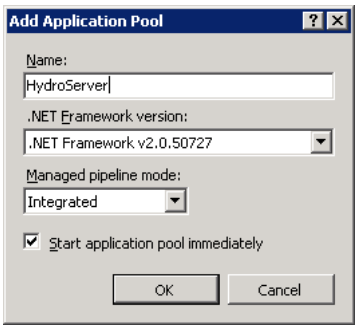
2. Open the Internet Information Services (IIS) Manager by clicking Start → Administrative Tools → Internet Information Services (IIS) Manager. The following window will appear.



3. Expand your server in the tree view at the left of the form by clicking on the plus sign next to its name. Then, expand the "Sites" element by clicking on the plus sign.
4. Since the ODMWDL was created in Microsoft Visual Studio 2008, we will first create an Application Pool for running it. In the tree view on the left, right click on "Application Pools" and select "Add Application Pool". The following window will appear.

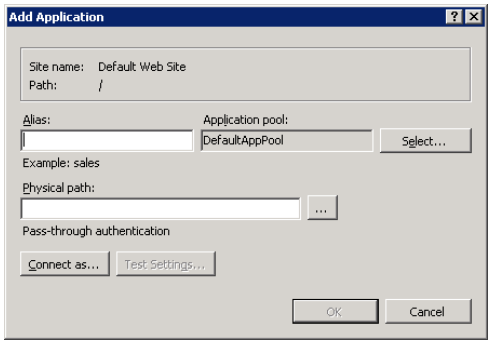


5. Create a name for the Application Pool in the "Name:" text box. For this example, we will call our Application Pool "HydroServer." Make sure that ".NET Framework v2.0.50727" is selected in the ".Net Framework version:" text box. From the "Managed pipeline mode:" drop down box, make sure that "Integrated" is selected. Ensure that the check box next to "Start application pool immediately" is checked. Your form should look like the following:



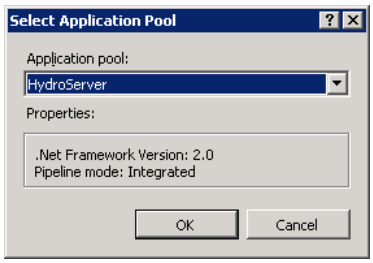
NOTE: In this example we are creating an Application Pool called "HydroServer." We have chosen the "Integrated" Managed pipeline mode because the ODMWDL was created using Visual Studio 2008. If we implement other HydroServer web applications (e.g., the HydroServer Capabilities Web Service and the HydroServer Time Series Analyst) that were developed in Visual Studio 2008, we can reuse this Application Pool for those applications.

6. For this example, we will implement the ODMWDL under the default website (see note below about setting ODMWDL up under a website other than the default website). Right click on the name of the website (in this example "Default Web Site") in the IIS tree view and select "Add Application" from the context menu. The following window will open:

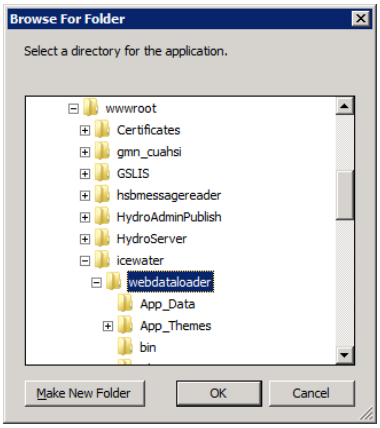


NOTE: ODMWDL was designed to be set up as an application that is subsidiary to a domain within IIS. For example, the URL for ODMWDL will be http://yourdomain/odmwdl/. The domain can be the name of the server in the case that you set ODMWDL up under the default website in IIS, or it can be a domain of your choosing.

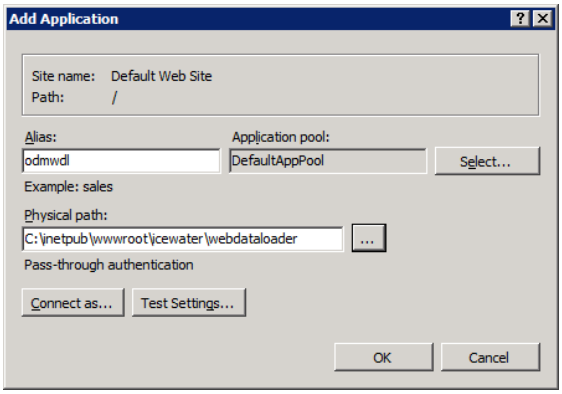
7. In the "Alias" text box, enter "odmwdl." Click the "Select" button next to the "Application pool:" box. In the "Select Application Pool" form that pops up, select the "HydroServer" application pool from the "Application pool" drop down list and then click "OK".



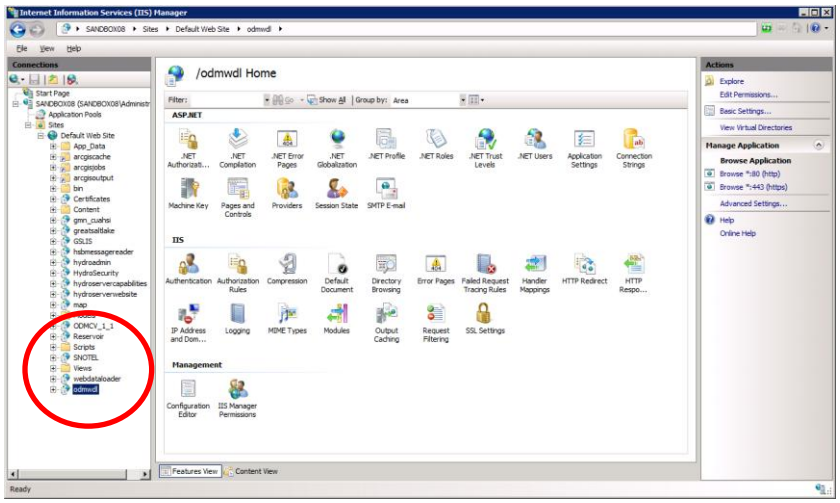
8. Click the navigate button next to the “Physical path:” box. The following window will open. Navigate to and select the folder where you extracted the TSA application files (e.g., “c:\inetpub\wwwroot\icewater\WebDataLoader” for this example). Then click the “OK” button.



9. Your “Add Application” form should now look something like the following. Click the “OK” button to complete this step.



10. You will now notice that an application called “odmwdl” has been created under your website in IIS.



NOTE: The process for setting up ODMWDL under a different website is the same as setting it up under the default website. However, you must first have created the website in IIS before you can do this. In the example above, since I implemented ODMWDL under the default website on a machine called "sandbox08.uwrl.usu.edu", my ODMWDL will have a URL of http://sandbox08.uwrl.usu.edu/odmwdl/. If I wanted my ODMWDL to have a different URL, I would first have to set up an appropriate domain name and then set up a website in IIS to handle that domain name. For example, I could register a domain name called "icewater.usu.edu" and have it pointed at this same machine. I would then create a website in IIS called "icewater" that would handle the "icewater.usu.edu" domain. Then, I would follow the steps above to set up ODMWDL under the "icewater" website in IIS. The path for my ODMWDL would then be http://icewater.usu.edu/odmwdl/. If you wish to register alternate domain names for your HydroServer, you will need to work with the individuals in charge of your IT infrastructure.

Congratulations! Your ODMWDL setup is now complete. You should be able to navigate to your ODMWDL in a web browser. For this example, the URL would be http://sandbox08.uwrl.usu.edu/odmwdl/.

3.0 Loading Data Using the ODM Web Data Loader

Loading data using the ODMWDL requires some knowledge of the structure of ODM. The purpose of this section is to provide the necessary background information that is needed to understand how the ODMWDL operates. Please refer to the ODM design specifications document (Tarboton et al., 2008), which is available on the CUAHSI HIS website http://his.cuahsi.org, for more a more detailed description of ODM and its features.

3.1 ODM Background

ODM is a relational data model. As such, there are relationships between most of the tables in ODM that must be maintained in order to protect the integrity of the data within the database. For example, there is a relationship between the Sites table and the DataValues table, with a primary key called SiteID in the Sites table that points to a foreign key called SiteID in the DataValues table. This linkage defines the relationship between observation values in the DataValues table and the monitoring site at which they were measured in the Sites table. Similar relationships exist for Variables, Methods, Samples, QualityControlLevels, Sources, etc. Because these relationships exist in ODM, DataValues cannot be loaded into the database until the records that define the site at which they were measured, the variable that they represent, the method used to make the measurements, etc., have been added to the associated tables within ODM.

Most of the tables within ODM have a primary key field that contains a unique integer identifier for each record in the table. For example, the Sites table has a primary key called SiteID, the Variables table has a primary key called VariableID, etc. Beginning with ODM Version 1.1, these primary key fields were converted to identity/autonumber fields to take advantage of the capability of commercial relational database management systems to protect the integrity of data in the database through automatic assignment of the primary key values. This means that the database manages the assignment of the primary key ID values rather than the user. ***In other words, you cannot import data into these primary key fields, the database creates the IDs automatically.***

Beginning with ODM Version 1.1, relationship also exist between the controlled vocabulary tables and the fields that they constrain. Because of this, terms cannot be loaded into fields that have a corresponding controlled vocabulary table unless those terms already exist in the controlled vocabulary tables. If appropriate terms do not exist in the controlled vocabulary tables for the data that you would like to load, please use the online forms that can be accessed through the CUAHSI HIS website (http://his.cuahsi.org) to request additions to the central ODM controlled vocabularies. Once new terms are added to the central controlled vocabulary system, they can be added to your local ODM database using the ODM Tools application.

3.2 Order of Operations

The primary key values (including controlled vocabulary terms) must be established before they can be used as foreign keys, so the order in which the data and its associated metadata are loaded matters. For example, a site must be loaded into the Sites table before data values measured at that site can be loaded into the DataValues table. When a site is loaded into the Sites table, the database automatically assigns it an integer SiteID. That SiteID is then used as the foreign key in the DataValues table to identify all of the observations collected at that site. Generally speaking, when data are loaded into ODM all of the metadata (i.e., sites, variables, methods, etc.) should be loaded *first*, followed by data values. With reference to the ODM schema diagram that can be found in the ODM Design Specifications document, you should start at the outside and work toward the DataValues table,

which is in the center of the diagram. In general, the following order of operations can be followed in loading data into ODM to avoid potential primary/foreign key constraint violations:

1. Load Sites
2. Load Variables
3. Load OffsetTypes
4. Load ISOMetadata
5. Load Sources
6. Load Methods
7. Load LabMethods
8. Load Samples
9. Load Qualifiers
10. Load QualityControlLevels
11. Load DataValues
12. Load GroupDescriptions
13. Load Groups
14. Load DerivedFrom
15. Load Categories

3.3 Methods for Loading Data

In general, there are two methods for loading data using the ODMWDL: 1) loading individual tables using separate input files for each ODM table, and 2) loading multiple or all of the ODM tables from a single input file. When using method 1, the ODMWDL is executed multiple times (once for each file to be loaded) using the simple table-based GUI, the command line interface, or the simple wizard. The above order should be followed in loading the data using method 1. When using method 2, the ODMWDL will manage the order of operations to make sure that everything gets loaded correctly. However, it should be noted that the ODMWDL is capable of loading the DataValues table with nearly any combination of or subset of the above tables (e.g., Sites and DataValues only, Variables and DataValues only, etc.). If you choose to use this advanced functionality, care must be taken to avoid potential errors. For example, if you wish to load Sites and DataValues from the same file, but that file does not include all of the Variable information, you must ensure that all of the Variable information has already been loaded into the database and that correct Variable references are set in the file you are trying to load (i.e., it contains a VariableID or VariableCode column to identify the variables associated with the data).

3.4 Creating Input Files

Appendix A of this document contains the templates for the input files that are accepted by the ODMWDL. The ODMWDL provides some flexibility in how you can specify the contents of the input data files to be loaded. For example, after you have loaded sites, you can use either the SiteID OR the SiteCode to identify the appropriate site for data values to be loaded. The input file templates specify the format of the input files and the alternatives that can be used in creating them. Input files must follow these templates or errors will occur. In general, files should only include the fields as specified in Appendix A and should not contain extra fields (i.e., do not include a DataValue column in your file if you only want to load the Sites table).

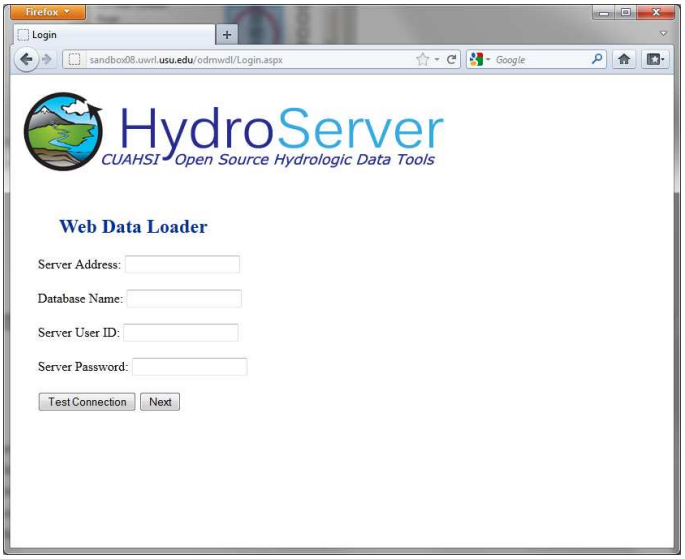
NOTE: Bulk data loading (i.e., loading multiple tables from a single file) can only be done when you are loading DataValues. Use the DataValues table template in Appendix A for bulk data loading tasks.

4.0 USING THE ODM WEB DATA LOADER

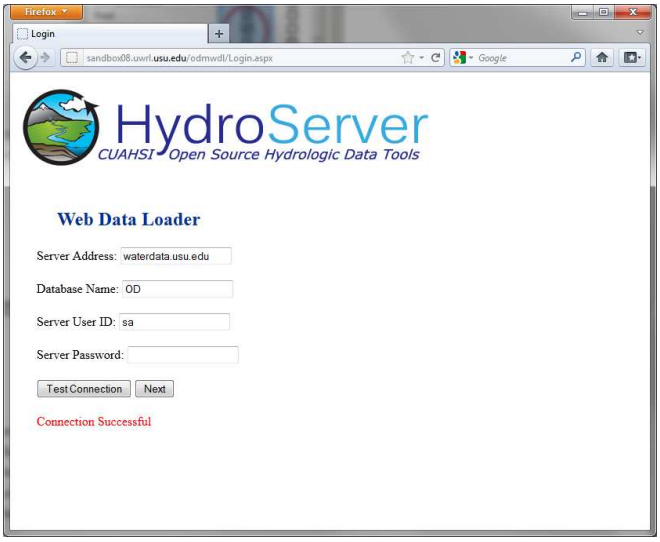
NOTE: The steps in this section assume that you already have an ODM database set up and running within an instance of Microsoft SQL Server 2005 or 2008. If you do not have SQL Server 2005 or 2008 or a working ODM database, please consult the information in the Installation Prerequisites section above for information on how to remedy this situation.

The first step in loading data is to create input files that match the templates listed in Appendix A and save them as comma delimited text files. Once you have done this, use the following steps to load data into your ODM database using the ODMWDL.

1. Open the ODMWDL by opening a web browser window and navigating to the URL of your web data loader. Your browser window will look something like the following:



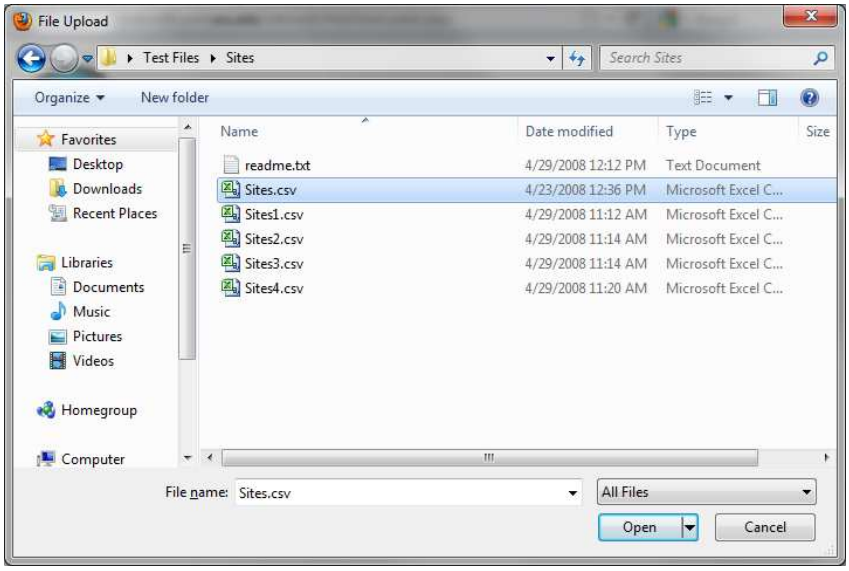
This is the form on which you will create a connection to the ODM database into which you want to load data. Enter an appropriate Server Address, Database Name, and SQL Server User Name and Password and then click on "Test Connection." If your connection to the database is successful, you will see a message at the bottom of the form.



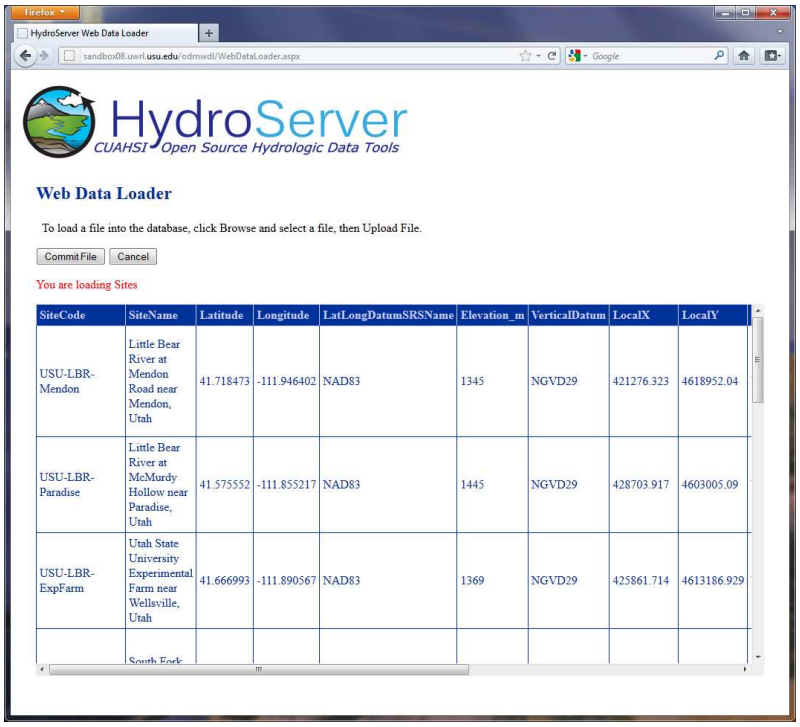
NOTE: The ODMWDL only supports SQL Server Authentication. It does not support Windows Authentication in SQL Server. You must have a valid SQL Server account with read and write access to the database that you would like to load data into. The ODMWDL can connect to any ODM database to which it can create a direct SQL connection.

2. Click the "Next" button to continue. You may have to re-enter your password. The following page will appear:

3. On this page, you will specify the file that you wish to load. Click on the "Browse" button and navigate to the location on your computer of the file that you want to load. In this example, we will load a Sites file that contains records for several monitoring sites in the Little Bear River.



Once you find the file to be loaded, click on the "Open" button. The ODMWDL will get the path for the file and load it into the text box on the page. Click the "Upload file" button. You will see a message stating that the file is being uploaded to the server, after which the following page will be displayed:



You will notice that the ODMWDL has recognized that we are loading a Sites file. The ODMWDL figures out what type of file you are loading from the header at the top of the file, so you do not have to tell it beforehand what you are trying to do. However, this does mean that you need to follow the input file templates given in Appendix A.

4. The simple, table-based interface of the data loader provides you with a preview of the data before it gets committed to the database. You can scroll through the data to make sure that it has been read correctly. To commit the data in your file to the database, click the "Commit File" button. If the data loader was successful in loading the data into the database, you will see a page similar to the following:

NOTE: If the load was unsuccessful, you will be provided with an error message that details why the data were rejected. Remember that the ODMWDL will not load any data into the database unless it can successfully load the entire file. If there are any errors found in the input file, the entire file will be rejected. This protects the integrity of the data in the database and avoids the potential problems caused by partial data loads.

5. If you wish to load another file to your database, click the "Add Another File to Current Database" button. If you want to load a data file to a different database, click the "Change Database" button to create a connection to a different database.

REFERENCES

Horsburgh, J. S., D. G. Tarboton, D. Maidment, and I. Zaslavsky (2008), A Relational Model for Environmental and Water Resources Data, Water Resources Research, Vol. 44, W05406, doi:10.1029/2007WR006392.

Tarboton, D.G., Horsburgh, J.S., and D.R. Maidment. 2008. CUAHSI Community Observations Data Model (ODM) Design Specifications Document: Version 1.1. http://his.cuahsi.org/odmdatabases.html.

APPENDIX A: INPUT FILE TEMPLATES FOR THE ODM WEB DATA LOADER

The general format for input to the ODMWDL is a single file containing a table with a one row header that uses ODM field names in the header, followed by the data in subsequent rows. The templates are such that the input data table format (i.e., the included columns) should either be identical to its destination table within ODM, or in expanded flat file format providing ancillary data associated with each data value sufficient to either load ancillary data tables or identify appropriate existing records in metadata tables. The ODMWDL identifies database fields from the input file header names, such that the order of columns in the input file does not matter. The ODMWDL identifies the contents of the input file and subsequently what the user is trying to load by parsing its header. The rules for identifying files by header information are given for each table below. If an input file fails to meet one of the rules specified below, an invalid file error will be returned.

Tables that can be loaded into ODM using the ODMWDL are listed below in alphabetical order. In the lists of field headers below, (R) indicates required and (O) indicates optional. Where field headers are listed in italics (for example see *SiteColumns* for the DataValues table) users have multiple options for specifying the content of the input file for those fields.

ODM Table: Categories

Identification Rule: ODMWDL will identify a Categories file by the appearance of CategoryDescription in the file header list without a DataValue field

Field Headers:

VariableColumns (M) EITHER one and only one of VariableID or VariableCode that corresponds to an existing Variables record in the Variables table.

DataValue (R)

CategoryDescription (R)

ODM Table: DataValues

Identification Rule: ODMWDL will identify a DataValues file by the appearance of DataValue in the field header list.

Field Headers:

DataValue (R)

ValueAccuracy (O)

LocalDateTime (R^1)

UTCOffset (R^1)

DateTimeUTC (R^1)

SiteColumns(M) EITHER one and only one of SiteID or SiteCode that corresponds to an existing Sites record in the Sites table, OR the required and optionally the optional columns from the Sites file below.

VariableColumns (M) EITHER one and only one of VariableID or VariableCode that corresponds to an existing Variables record in the Variables table, OR the fields listed for the Variables file below.

OffSetValue (O)

OffsetTypeColumns (O) EITHER OffsetTypeID that corresponds to an existing OffsetTypes record, OR the fields listed for the OffsetTypes file below.

CensorCode (R) – Must contain an existing term from the CensorCodeCV table.

QualifierColumns (O) EITHER QualifierID that corresponds to an existing Qualifiers record in the Qualifiers table, OR the fields listed for the Qualifiers file below.
MethodColumns (R) EITHER MethodID that corresponds to an existing Methods record in the Methods table, OR the fields listed for the Methods file below.
SourceColumns (R) EITHER SourceID that corresponds to an existing Sources record in the Sources table, OR the fields listed for the Sources file below.
SampleColumns (O) EITHER SampleID that corresponds to an existing Samples record in the Samples table, OR the fields listed for the Samples file below.
DerivedFromID (O)
QualityControlLevelColumns (R) EITHER QualityControlLevelID that corresponds to an existing record in the QualityControlLevels table, OR the fields listed for QualityControlLevels file below.
GroupDescription (O) If this matches an existing GroupDescription the corresponding GroupID and ValueID will be added to the Groups table. If this is new, a new GroupDescriptions record will be added and the corresponding IDs added to the Groups table.

[1]**Note:** Only two of LocalDateTime, UTCOffset and DateTimeUTC are required. The third will be calculated from the other two.

ODM Table: DerivedFrom

Identification Rule: ODMWDL will identify a DerivedFrom file by the appearance of DerivedFromID and ValueID in the field header list without any other fields (requires that DataValues have already been populated)
Field Headers:
DerivedFromID (R)
ValueID (R)

ODM Table: Groups

Identification Rule: ODMWDL will identify by a Groups file by the appearance of GroupID and ValueID in the field header list without any other fields (requires that GroupDescriptions and DataValues have already been populated)
Field Headers:
GroupID (R)
ValueID (R)

ODM Table: GroupDescriptions

Identification Rule: ODMWDL will identify a GroupDescriptions file by the appearance of GroupDescription in the field header list without a DataValue field.
Field Headers:
GroupDescription (R)

ODM Table: ISOMetadata

Identification Rule: ODMWDL will identify an ISOMetadata file by the appearance of TopicCategory in the field header list without DataValue and without Organization.

Field Headers:
TopicCategory (R) – This must contain a valid term from the TopicCategoryCV table.
Title (R)
Abstract (R)
ProfileVersion (R)
MetadataLink (O)

ODM Table: LabMethods

Identification Rule: ODMWDL will identify a LabMethods file by the appearance of LabName in the field header list without DataValue and without SampleType.
Field Headers:
LabName (R)
LabOrganization (R)
LabMethodName (R)
LabMethodDescription (R)
LabmethodLink (O)

ODM Table: Methods

Identification Rule: ODMWDL will identify a Methods file by the appearance of MethodDescription in the field header list without DataValue.
Field Headers:
MethodDescription (R)
MethodLink (O)

ODM Table: OffsetTypes

Identification Rule: ODMWDL will identify an OffsetTypes file by the appearance of OffsetDescription in the field header list without DataValue.
Field Headers:
OffsetUnitsColumn (R) OffsetUnitsID (referring to UnitsID in the Units table) or OffsetUnitsName (referring to UnitsName in the Units Table). One and only one of these columns should be present matching to an existing record in the Units table.
OffsetDescription (R)

ODM Table: Qualifiers

Identification Rule: ODMWDL will identify a Qualifiers file by the appearance of QualifierDescription in the field header list without DataValue.
Field Headers:
QualifierCode (O)
QualifierDescription (R)

ODM Table: QualityControlLevels

Identification Rule: ODMWDL will identify a QualityControlLevels file by the appearance of QualityControlLevelCode in the field header list without DataValue.
Field Headers:
QualityControlLevelCode (R)
Definition (R)
Explanation (R)

ODM Table: Samples

Identification Rule: ODMWDL will identify a Samples file by the appearance of SampleType in the field header list without DataValue.
Field Headers:
SampleType (R) – This must contain a valid Term from the SampleTypeCV table.
LabSampleCode (R)
LabMethodColumns (R) EITHER LabMethodID that corresponds to an existing record in the LabMethods table OR the columns listed for the LabMethods table below.

ODM Table: Sites

Identification Rule: ODMWDL will identify a Sites file by the appearance of SiteName in the field header list without the appearance of DataValue.
Field Headers:
SiteCode (R)
SiteName (R)
Latitude (R)
Longitude(R)
LatLongDatumColumn (R) LatLongDatumId (referring to SpatialReferenceID in the SpatialReferences table) or LatLongDatumSRSID (referring to SRSID in the SpatialReferences table) or LatLongDatumSRSName (referring to SRSName in the SpatialReferences table). One and only one of these is required and should be used to identify the corresponding record in the SpatialReferences controlled vocabulary table upon loading.
Elevation_m (O)
VerticalDatum (O) – If used, this must contain a valid term from the VerticalDatumCV table.
LocalX (O)
LocalY(O)
LocalProjectionColumn (O) LocalProjectionID (referring to SpatialReferenceID in the SpatialReferences table) or LocalProjectionSRSID (referring to SRSID in the SpatialReferences table) or LocalProjectionSRSName (referring to SRSName in the SpatialReferences table) (O). One and only one of these is required if LocalX and LocalY are specified and should be used to identify the corresponding record in the SpatialReferences controlled vocabulary table upon loading.
PosAccuracy_m (O)
SiteState (O)
County (O)
Comments (O)
SiteType (O) (Only for ODM 1.1.1 Databases)

NOTE: SiteState is different than the field name in ODM. The use of SiteState in the input file template is necessary to facilitate bulk data loading because there is also a State field in the Sources table. Additionally, you will notice there there are no LatLongDatumSRSID, LatLongDatumSRSName, LocalProjectionSRSID, or LocalProjectionSRSName fields in ODM. These fields exist only in the template and reference the SRSID and SRSName fields in the SpatialReferences table to provide alternate inputs to the Sites table.

NOTE: SiteType is only supported for ODM 1.1.1 databases.

ODM Table: Sources

Identification Rule: ODMWDL will identify a Sources file by the appearance of Organization in the field header list without DataValue.

Field Headers:

Organization (R)
SourceDescription (R)
SourceLink (O)
ContactName (R)
Phone (R)
Email (R)
Address (R)
City (R)
SourceState (R)
ZipCode (R)
MetadataColumns (R) EITHER MetadataID that corresponds to an existing ISOMetadata record OR the columns listed for the ISOMetadata table below.
Citation (R)

NOTE: SourceState is different than the field name in ODM. The use of SourceState in the input file template is necessary to facilitate bulk data loading because there is also a State field in the Sites table.

ODM Table: Variables

Identification Rule: ODMWDL will identify a Variables file by the appearance of VariableName in the field header list without the appearance of DataValue.

Field Headers:

VariableCode (R)
VariableName (R) – This must contain a valid term from the VariableNameCV table.
Speciation (R) – This must contain a valid term from the SpeciationCV table.
VariableUnitsColumn (R) VariableUnitsID (referring to UnitsID in the Units table) or VariableUnitsName (referring to UnitsName in the Units Table). One and only one of these columns should be present matching to an existing record in the Units table.
SampleMedium (R) – This must contain a valid term from the SampleMediumCV table.
ValueType (R) – This must contain a valid term from the ValueTypeCV table.
IsRegular (R)

TimeSupport (R)
TimeUnitsColumn (R) TimeUnitsID (referring to UnitsID in the Units table) or TimeUnitsName (referring to UnitsName in the Units Table). One and only one of these columns should be present matching to an existing record in the Units table
DataType (R) – This must contain a valid term from the DataTypeCV table.
GeneralCategory (R) – This must contain a valid term from the GeneralCategoryCV table.
NoDataValue (R)