HydroServer Web Data Loader
Functional Specifications

Jeffery S. Horsburgh[1] and David G. Tarboton[2]

10/11/2010

1. Introduction

The Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) is an organization representing over 100 universities that is supported by the National Science Foundation to develop infrastructure and services for the advancement of hydrologic science in the United States. CUAHSI's mission has several components, one of which is the development of a Hydrologic Information System (HIS) to assemble and synthesize hydrologic data to support hydrologic science development. The CUAHSI HIS is being developed as a geographically distributed network of hydrologic data sources and functions that are integrated using web services so that they function as a connected whole. One objective of HIS is establishing the cyberinfrastructure foundation, or digital environment required to support experimental watersheds or hydrologic and environmental observatories.

The CUAHSI HIS project has developed a standard software stack called HydroServer that enables researchers to publish observational data. One component of the HydroServer software stack is a database schema called the Observations Data Model (ODM) for storing point observations in a relational database. ODM is currently being implemented at a number of experimental watersheds, test bed project locations, and hydrologic or environmental observatories throughout the country as a mechanism for publication of individual investigator data and for registering these data with the CUAHSI National HIS. At many of these locations, researchers are working collaboratively to collect data, and in some cases institutional networking and information technology policies can constrain access to HydroServers and ODM databases. For example, one institution's firewall settings may preclude someone from outside that institution from connecting to an ODM database to load data. To address this situation, this document specifies the functional requirements for a web-based data loader for ODM (hereafter referred to as ODMWDL) that will overcome this potential limitation to collaborative hosting of a HydroServer and ODM databases.

2. Primary Use Case

Broadly the web based data loader will be a web forms based version of the ODM data loader. The functionality required is illustrated by the following use case:

1. A user prepares data for loading into an ODM database by following the guidance given for ODM data loader. This results in one or more ODM Data Loader input files
2. The user authenticates on a web loading section of the HydroServer web page.

[1] Utah Water Research Laboratory, Utah State University, 8200 Old Main Hill, Logan, UT, 84322-8200, (435) 797-2946, jeff.horsburgh@usu.edu.
[2] Utah Water Research Laboratory, Utah State University, 8200 Old Main Hill, Logan, UT 84322-8200, (435) 797-3172, david.tarboton@usu.edu

3. The user identifies (e.g., from a drop down list) which ODM database the data is to be loaded into.
4. The user selects a browse button to identify an input file on his/her local system to be loaded.
5. The user clicks a button to initiate the upload of the input data file.
6. The system uploads the file and parses it as it would for the ODM Data Loader, displaying the data to be loaded in a table similar to ODM Data Loader. If the input data file is large, only a subset of the records in the input file will be shown in the preview.
7. If the data is complete and sufficient to be input to the database, a "Commit" button will be enabled. If the data is incomplete or insufficient for loading, for any reason, messages indicating the problem will be displayed, and the user will be offered the option to browse to a corrected input file (i.e., return to step 4) and try again.
8. The user clicks the "Commit" button, at which point the data is loaded in to the ODM database. The series catalog table is updated and the data are made available through all functionality supported by HydroServer (e.g., Time Series Analyst, WaterOneFlow Web Services, etc.)

3. Features and Functional Requirements

The general concept behind ODMWDL is that it provides a simple, web-based interface for loading data into an ODM database from any computer that has a web browser and an Internet connection. It will overcome many common limitations in institutional firewalls that limit users from making direct SQL connections to ODM databases from outside of an institution. Where possible, the ODMWDL will use the code base from the existing ODMDL application. The functionality of the web-based version will be similar to that of the existing ODMDL.

3.1. Authentication and Authorization

Since loading data requires write permissions on an ODM database, the web data loader will require users to be authenticated and authorized before they can load data. This will require development of a system for users to authenticate themselves and for a HydroServer administrator to authorize particular users to load data using ODMWDL. This will be part of the broader authentication and access control functionality being designed for HydroServer. Since the HydroServer Security and Access control system is in the process of being developed, an initial simple approach for authentication and authorization may be implemented so that a HydroServer Adiminstrator can manage users and assign permissions for web loading of data to one, some, or all ODM databases on a HydroServer.

3.2. Input Files

ODMWDL will accept as input data in table format (comma or tab separated) that is sufficient that it can be loaded into ODM without violating any ODM constraints. Input files will have a one row header that uses ODM field names in the header, followed by the data in subsequent rows. Similar to ODMDL, ODMWDL will support loading of individual ODM data tables (e.g., Sites, Variables, Methods, etc.). The ODMWDL interface should be such that a user does not need to designate the table into which data is to be loaded. Based on the header information in a file sent to ODMWDL, the type of content in the file should be automatically determined and one of the following operations used to parse and input the file:

1. Import Sites – Import data from the Sites template into the Sites table.

2. Import Variables – Import data from the Variables template into the Variables table.
3. Import Sources – Import data from the Sources template into the Sources table.
4. Import Methods – Import data from the Methods template into the Methods table.
5. Import LabMethods – Import data from the LabMethods template into the LabMethods table.
6. Import Samples – Import data from the Samples template into the Samples table.
7. Import Qualifiers – Import data from the Qualifiers template into the Qualifiers table.
8. Import OffsetTypes – Import data from the OffsetTypes template into the OffsetTypes table.
9. Import DataValues – Import data from the DataValues template into the DataValues table.
10. Import ISOMetadata – Import data from the ISOMetadata template into the ISOMetadata table
11. Import Categories – Import data from the Categories template into the Categories table
12. Import Groups – Import data from the Groups template into the Groups table
13. Import GroupDescriptions – Import data from the GroupDescriptions template into the GroupDescriptions table
14. Import DerivedFrom – Import data from the DerivedFrom template into the DerivedFrom table
15. Import QualityControlLevels – Import data from the QualityControlLevels template into the QualityControlLevels table.

Appendix A provides format templates for each of the ODMDL data import tasks and lists the required fields.

3.3. Accessing ODM Databases

Users that wish to load data into an ODM database on a HydroServer will log into the ODMWDL. They will then be presented with a list of ODM databases that they have been given permission to load data into. Users will select the database into which they want to load data from the list.

3.4. Log File Generation

ODMWDL will write information to a text log file when it successfully loads data into an ODM database. Information in the log file will include dates that the loader was executed, information about the file(s) being loaded, success or failure in loading the intended file(s), and any specific error information needed to evaluate data loading successes or failures. ODMWDL will also report similar information to the web page that the user sees to report successful data loads or error information in the case of execution failure.

3.5. Data Validation, Integrity Checks, and Transaction Management

ODMWDL will implement two levels of validation on data that are to be loaded to ensure that the integrity of the data in an ODM database is maintained. First, upon reading an input file and displaying a preview of the file on a web page for the user to see, it will be parsed and checked for consistency with

ODM requirements and constraints. This includes checking data types, required fields, fields that cannot be null, fields that do not allow special characters, fields that must conform to controlled vocabularies, etc. Once the file to be loaded has been validated at this level, it will be passed to the database for loading. The second level of validation occurs when ODMWDL tries to insert the data into the database (i.e., the ODM database will apply all of its constraints). If the data violate any of the constraints of ODM, ODMWDL will capture the error and report it back to the user with a suggestion on how to fix it if possible. ODMWDL will assume that each file to be loaded is a single transaction and that the entire file must be loaded or none of the file is loaded. Efforts will be made to report the line number of the file at which invalid data or errors occur so that users can correct potential errors in the input files. ODMWDL will check to make sure that records added to each table within ODM are unique so that duplicates are not loaded into the database.

4. Technical Requirements

4.1. Development Environment and Source Code

ODMWDL will be built as a web application in the Microsoft Visual Studio 2008 or 2010 development environment. The application will use ASP.NET, and the language of the application will be C# or Visual Basic, depending on the degree to which the existing code base can be reused. ODMWDL and its source code will be made freely available according to the CUAHSI HIS software policy on the HydroServer CodePlex website (http://hydroserver.codeplex.com).

4.2. Testing

The functionality of the ODMWDL will be tested for compliance with the Microsoft Internet Explorer, Mozilla Firefox, and Apple Safari Internet Browsers.

4.3. Support for Data Files

ODMWDL will support input data files in comma- or tab-delimited text format. Appendix A provides templates indicating required fields for each of the input tables supported by ODMWDL. ODMWDL will automatically determine which file type is being loaded from its extension and contents according to the rules given in Appendix A.

4.4. User Interface Requirements

ODMWDL will be a web-based application that runs in a web browser. It will not require any software or components to be installed other than the operating system and the web browser.

4.5. Installation and Configuration

ODMWDL will be delivered via a zipped application directory that can be distributed from the CUAHSI HIS HydroServer CodePlex website (http://hydroserver.codeplex.com). A software manual will be provided with instructions on how to install and configure the application on a HydroServer.

Appendix A
ODMWDL Input File Templates

The general format for these templates 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. ODMWDL will identify database fields from the input file header names, such that the order of columns in the input file does not matter. ODMWDL will identify the contents of the input file by parsing its header. The rules for identifying files by header information are given below for each table. If an input file fails to meet one of the rules specified below, an invalid file error will be returned.

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: DataValues
Identification Rule: ODMDL will identify a datavalues file by the appearance of DataValue in the field header list.
Field Headers:

- DataValue (R)
- ValueAccuracy (O)
- LocalDateTime (R1)
- UTCOffset (R1)
- DateTimeUTC (R1)
- 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)
- 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 should be added to the Groups table. If this is new, a new GroupDescriptions record should be added and the corresponding IDs added to the Groups table.

Notes:

1. Only two of LocalDateTime, UTCOffset and DateTimeUTC are required. The third may be calculated from the other two.
2. Duplicate data values are permitted because they may actually be valid in the case of multiple replicates of a measurement.

ODM Table: Sites
Identification Rule: ODMDL should identify a sites file by the appearance of SiteName without the appearance of DataValue in the header list.
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)
- 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)

ODM Table: OffsetTypes
Identification Rule: Identify the OffsetTypes file by the appearance of OffsetDescription without the appearance of DataValue in the header list.
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: Variables

Identification Rule: Identify the Variables file by the appearance of VariableName without the appearance of DataValue in the header list.
Field Headers:

- VariableCode (R)
- VariableName (R)
- Speciation (R)
- 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)
- ValueType (R)
- 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)
- GeneralCategory (R)
- NoDataValue (R)

ODM Table: Sources (Check specification for required or optional)
Identification Rule: Identify by Organization 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)

ODM Table: Methods
Identification Rule: Identify by MethodDescription without DataValue
Field Headers:

- MethodDescription (R)
- MethodLink (O)

ODM Table: Samples
Identification Rule: Identify by SampleType without DataValue
Field Headers:

- SampleType (R)
- 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: LabMethods
Identification Rule: Identify by LabName without DataValue and without SampleType
Field Headers:
- LabName (R)
- LabOrganization (R)
- LabMethodName (R)
- LabMethodDescription (R)
- LabmethodLink (O)

ODM Table: Qualifiers
Identification Rule: Identify by QualifierDescription without DataValue
Field Headers:
- QualifierCode (O)
- QualifierDescription (R)

ODM Table: ISOMetadata
Identification Rule: Identify by TopicCategory without DataValue or Organization
Field Headers:
- TopicCategory (R)
- Title (R)
- Abstract (R)
- ProfileVersion (R)
- MetadataLink (O)

ODM Table: QualityControlLevels
Identification Rule: Identify by QualityControlLevelCode without DataValue field.
Field Headers:
- QualityControlLevelCode (R)
- Definition (R)
- Explanation (R)

ODM Table: Categories
Identification Rule: Identify by CategoryDescription without LocalDateTime, DateTimeUTC, or UTCOffset fields
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)

Note that we need special code to handle the loading of categorical data. I suggest allowing a DataValue of "C" in the input datavalues table which indicates to the loader that the DataValue is categorical. The corresponding variable should have categorical datatype (that should be checked or created as categorical if it is being created). CategoryDescription should then be an allowed column in the

DataValues file and the loader should match the category description and variableID entries to assign the corresponding numeric DataValue, or if not matched, create a new categorical mapping.

ODM Table: Groups
Identification Rule: Identify by GroupID and ValueID fields without any other fields (requires that GroupDescriptions and DataValues have already been populated)
Field Headers:

- GroupID (R)
- ValueID (R)

ODM Table: GroupDescriptions
Identification Rule: Identify by GroupDescription without DataValue field
Field Headers:

- GroupDescription (R)

ODM Table: DerivedFrom
Identification Rule: Identify by DerivedFromID and ValueID fields without any other fields (requires that DataValues have already been populated)
Field Headers:

- DerivedFromID (R)
- ValueID (R)