ODM Data Loader
Functional Specifications

June 3, 2008

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

Introduction

The CUAHSI Hydrologic Information System (HIS) Project is developing information technology infrastructure to support hydrologic science. One of the components of the HIS is a point Observations Data Model (ODM), which is a relational database schema that was designed for storing time series data. The purpose of ODM is to provide a framework for optimizing data storage and retrieval for integrated analysis of information collected by multiple investigators. This document describes the functional specifications for the ODM Data Loader. This application will provide users with the ability to load data into an instance of the CUAHSI ODM. The ODM Data loader will be structured so that it can be implemented either through a Graphical User Interface (GUI) or via a command line so that it can be run in batch mode or integrated with existing applications such as the ODM Tools application. The ODM Data Loader will be implemented in such a way that it can be deployed on individual users' machines so that they can load data into a local or remote (i.e., server based) ODM Database.

ODM Data Loader

This document describes the functional specifications for a software application that will be called ODM Data Loader, which will subsequently be referred to as "ODMDL." ODMDL is an existing application that was developed at the San Diego Supercomputer Center (SDSC) and is being modified to meet the needs of the CUAHSI HIS project. This is being done in such a way that it is consistent with and compatible with the CUAHSI HIS ODM Version 1.1. The following sections describe the major features and functionality that will be included in the ODMDL.

Features and Functional Requirements

The general concept behind ODMDL is that it should accept as input data in table format (Excel, CSV, or tab separated) that is sufficient that it can be loaded into ODM without violating any ODM constraints. Tables should have a one row header that uses ODM field names in the header, followed by the data in subsequent rows. Where possible, the ODMDL application will use the existing code base for the ODMDL application developed at SDSC. The functionality of the enhanced version (i.e., the version that will result from these functional specifications) will

[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

include what has already been implemented in the existing ODMDL, with a few enhancements. This functionality includes:

1. Bulk data loading – This is the current functionality of the ODMDL. It requires all of the information needed for data to be loaded in a single input file. The enhanced version of ODMDL will support this functionality.
2. Loading of individual ODM data tables – The enhanced ODMDL will support the independent loading of Sites, Variables, Methods, etc.
3. Sequential data loading – The enhanced ODMDL will provide a Wizard like GUI for sequential data loading. Users will first import Sites, then Variables, then Methods, etc. The last step will be to populate the actual data values. This Wizard will be constructed in such a way that it can be run as a stand-alone application or launched from other applications such as ODM Tools.
4. Command line Interface and batch data loading – The enhanced ODMDL will support command line execution with command line arguments that will enable batch data loading of files.

Establishing a Connection to an ODM Database

ODMDL will provide functionality for connecting to a local or remote ODM database implemented in Microsoft SQL Server 2005. Functionality will be provided for using Windows or SQL Server authentication in the database connection.

Bulk Data Loading

ODMDL will support the current functionality of loading data from a single file. This functionality will be supported through both a GUI and through a command line executable. The input file must contain all of the required information needed to load data into the content tables in ODM. A template for the bulk data loading input file is included in Appendix A of this document (see the template for the DataValues table).

Loading of Individual ODM Data Tables

ODMDL will support loading data from input files into individual tables in the ODM. This functionality will be supported through both a GUI and through a command line executable. The following individual data import tasks will be supported:

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.

Sequential Data Loading

ODMDL will allow users to load data sequentially. This will be in the form of a Wizard that guides users through loading the data in the correct order such that the constraints in the database will not be violated. This wizard will allow import from files as well as implementing functionality from the ODM Streaming Data Loader that will allow users to create new sites, variables, and other metadata by typing information into a form. This Wizard will be targeted at users that want to be guided through the steps of data loading in the correct sequence.

Command Line Interface

The ODMDL executable file will support a command line interface for loading data from files. This will enable scripting (i.e., scheduled data loads via computer code) and batch data loading of many files as described in the following section. When executing the ODMDL via the command line, the following arguments will be supported:

-connection [connection string](required) **–** This will allow the user to specify a valid connection string for a destination ODM database.

-filepath [file path string] (required) **–** This will allow the user to specify the file that is to be loaded.

The ODMDL executable will determine the format of the file from its extension and contents and will determine what is to be loaded from the contents of the file given the rules in Appendix A. Invalid input files will result in an invalid file error.

Batch Data Loading

ODMDL will support automated batch data loading. A single execution of the ODMDL executable will load a single input file. Batch data loading will require multiple executions of the ODMDL executable to load multiple files. This can be done either using a batch file that specifies repeated executions with all of the required command line arguments for each execution, or through code that manages repeated execution of the ODMDL executable.

Log File Generation

ODMDL will write information to a text log file when it executes both from the command line and from the simple table based GUI. 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 failures. ODMDL will also report similar information to the command line console when executing from a command prompt and to a text display on the simple ODMDL GUI.

Data Validation, Integrity Checks, and Transaction Management

ODMDL will implement two levels of validation on data that are to be loaded to ensure that the integrity of the data is maintained. First, upon reading the file and loading it into ODMDL, 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 ODMDL 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, ODMDL will capture the error and report it back to the user. ODMDL 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 occurs so that users can correct potential errors in the input files. ODMDL will check to make sure that records added to each table within ODM are unique so that duplicates are not loaded into the database.

Technical Requirements

The following sections detail specific technical requirements for the ODMDL:

Development Environment and Source Code

ODMDL will be built as a Windows application in the Microsoft Visual Studio 2005 development environment. The language of the application will be C# or Visual Basic, depending on the degree to which the existing code base can be reused. The ODMDL application and its source code will be made freely available according to the CUAHSI HIS software policy.

Operating System Support

ODMDL will be tested for use on Microsoft Windows XP and Microsoft Windows 2003 server (32-Bit Version) with Version 2.0 of the Microsoft .Net Framework.

Database Support

ODMDL will be designed to connect directly to an instance of the CUAHSI HIS ODM Version 1.1 implemented in Microsoft SQL Server 2005 (including SQL Server 2005 Express). The GUI for the application will provide the user with a simple interface for creating a connection to the database, including server and authentication information. Executing ODMDL via the command line will require a valid connection string as an argument. ODMDL will support connection to either local or remote database servers (i.e., users will be able to install the ODM SDL application on their own PC and connect to either a local or remote server running Microsoft SQL Server 2005). Documentation in the form of a User's Manual will be provided with the application to support users in creating a connection to the database(s), including server and authentication information.

Support for Data Files

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

User Interface Requirements

ODMDL will be a Microsoft Windows-based application. It will have a command line interface that can be used for batch data loading as well as and a GUI that will enable interactive data loading.

Installation and Configuration

ODMDL will be delivered via an executable installation file that can be distributed via compact disk or downloaded from the ODM website at http://water.usu.edu/cuahsi/odm/. The software installation will install all of the necessary components and files for the ODMDL application to work. It should be noted, however, that the software installation for ODMDL will install the software application, but it is left to the user to create an appropriate ODM database within Microsoft SQL Server 2005 for the ODMDL application to attach to.

Appendix A
ODMDL 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. ODMDL will identify database fields from the input file header names, such that the order of columns in the input file does not matter. ODMDL 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 (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)
- *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. (In future versions of ODM we will consider having a replicate indicator – like HarmoniRib)

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 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)

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)