ODM2 Admin Pilot Final Report
Miguel Leon, Rick Hooper, and Martin Seul

Introduction

During the past year, CUAHSI has executed a pilot project to explore the utility of a software package, ODM2Admin, for data management as a possible expansion of the services that CUAHSI provides to the academic research community. This report documents the findings from this pilot for the consideration of the Informatics Standing Committee. The project was executed as a subaward of $50,000 to the University of Pennsylvania to fund Miguel Leon, data manager for the Luquillo CZO, to adapt ODM2Admin to four additional

The rationale for evaluating this particular packages includes the following factors:

- ODM2Admin can be customized with Python scripts, which is a widely used programming language within the the CUAHSI community
- ODM2Admin is built in the Django environment, the same as Hydroshare, so the software environment is familiar to CUAHSI software engineers and can be readily supported by CUAHSI
- ODM2Admin was developed for the Luquillo CZO and has many of the basic functions that would be used by field sites, such as ingestion of data logger files
- ODM2Admin is built on the ODM2 data schema that permits managing both sensor and sample data and has extensions to manage information about field instrumentation, such as replacement of sensors, calibration data, etc.

The reasons for providing field data management services include the following:

- Scientists are deploying more sensors and dealing with ever larger data volumes
- Many ad hoc data management systems are being developed by scientists using resources they have at hand (such as data management software provided by sensor or data logger manufacturers) with customization to accommodate local workflow needs. This is inefficient, labor intensive, and frequently fails to achieve basic data management best practices. However, previous NSF-funded projects that attempted to develop field data management systems have uniformly failed. Generally, there are insufficient funds to develop a customizable system and to provide adequate user support for adoption of the system by a large community.
- Data publication now occurs as a separate step at the end of a project and is burdensome. The pilot includes a “one-button” publication system to allow users to select data to report data to CUAHSI HIS or Hydroshare.
- Providing ODM2Admin in a cloud environment removes dependence on local hardware and local firewall policies. The marginal cost of provisioning an additional ODM2Admin
service is small so the service can be funded either as a free service (using CUAHSI resources) or with a small fee to active projects for establishing the data service.

The pilot project engaged four different projects representing a range of scale and sophistication in data management. The four projects are

1. A second CZO site (Catalina/Jemez) to test transferability of the system to another field project with a data manager at University of Arizona
2. Dry Creek, a long-term field site, used for education and research at Boise State
3. An informal network of stream gaging sites in the Au Sable and Manistee watersheds operated by Michigan State University
4. CZIMEA, a cross-CZO synthesis project on microbial DNA sequencing where ODM2Admin manages the soil and site information where team members are sophisticated software developers.

This report describes the basic features of ODM2Admin and the experience of these four groups in using this system. The key question we are trying to answer is the degree of support each group requires to successfully implement and use ODM2Admin as their data management system, as a function of the complexity of the field project and the data management capabilities of the local project participants.

Background

ODM2 Admin is an application for site-level data management of environmental observations using Observation Data Model 2 (ODM2). The application was designed for management of observations generated by the Luquillo Critical Zone Observatory (LCZO), an NSF-funded Observatory established in 2009, located in northeastern Puerto Rico. The LCZO is one of NSF funded nine critical zone observatories, it features a dynamic tropical rainforest, with a long history of scientific inquiry and natural resource management. The LCZO manages in situ stream, groundwater, atmospheric, and soil sensors at 53 locations throughout the Luquillo Mountains. The LCZO also supports a number of field campaigns where samples of soils, rocks, and water are taken and analyzed in a laboratory. The results of these analyzes inform LCZO researchers on rates of weathering, sources and sinks of carbon, nitrogen and phosphorus, riparian zone dynamics among many other subjects of inquiry.

Managing the diverse range of data and associated metadata generated from studies of the LCZO requires a robust information model. To meet this requirement the LCZO adopted Observation Data Model 2 (ODM2) which serves as the foundation of ODM2 Admin. ODM2 is designed to manage data generated from discreet earth observations across many disciplines such as those made through the CZO program. Once the LCZO began using ODM2 it became clear, that due to the many interrelated entities in ODM2, an interface was needed to more easily understand the information model. Creating records in an ODM2 database requires associating entities through foreign keys and often those entities themselves are related to
additional entities. Keeping track of these relationships, using only a tool like SQL, is very difficult and time consuming. As a result, the LCZO began developing ODM2 Admin to help manage data in this complex but robust information model.

ODM2 Admin uses PostgreSQL, the Django web framework, and javascript APIs to create web forms for user friendly representations of ODM2 entities and their relationships. The Django web framework has a python backend for business logic and database interaction with html templates to generate the user interface. These technologies were chosen to be open source and consistent with other software developed as part of the ODM2 software ecosystem [https://github.com/ODM2/odm2-software-ecosystem](https://github.com/ODM2/odm2-software-ecosystem). These forms feature fields using drop down lists, type ahead fields and related entities via in-line forms. Many of the entities in ODM2 are meant to be managed via a community driven controlled vocabulary. Such as system has been established here [http://vocabulary.odm2.org/](http://vocabulary.odm2.org/). The ODM2 Admin web forms, with links into the controlled vocabulary system, help facilitate understanding of the terms contained in the controlled vocabulary. Time series and sample based results are mapped via leaflet and can be graphed via highcharts, both common widgets used in websites. Time series data from sensors can be loaded via web form initiated Django commands or via the command line. Access to Django and custom ODM2 Admin commands enable users to manage their ODM2 database and connect it with networked sensors. See the appendix of this report for further data.

ODM2 Admin is integrated with Hydroshare for publication of datasets. When a user wants to publish data they can add results, produced from time series data or specimen analysis, to datasets. Datasets can then be exported to Hydroshare, exporting will generate an SQLite version of the results and uploaded it to Hydroshare via the Hydroshare REST API along with associated metadata. Hydroshare fields will be populated from the dataset and all associated metadata will persist in the uploaded SQLite ODM2 database. ODM2 databases can also be registered with CUAHSI HIS for display on data.cuahsi.org using WOFpy, part of the ODM2 software ecosystem, both of these methods for data interchange are still being tested but should be ready soon.

**Approach**

Through the ODM2 Admin pilot project ODM2 Admin was evaluated to see how effectively other sites and research organizations could use ODM2 with ODM2 Admin. Four sites were engaged through a series of meetings in which ODM2 Admin was demonstrated, data managers and PIs from each site described the data they collect, and we discussed how their data could be managed in ODM2. The data managers at each site have different backgrounds and skills levels in managing data with databases.

The Dry Creek experimental forest, managed by Boise State, has a number of telemetered weather stations, stream gages, and soil sensors. Data logger files from these sites are automatically uploaded to an FTP site. Many of the time series in the data logger files were
documented in readme files on the FTP site or were described by the data manager in excel spreadsheets, this information then was loaded into an ODM2 database using python scripts and the Django Object Relational Mapper (ORM). Using ODM2 Admin with the Django ORM allows a user generate ODM2 database records with python scripts or commands. Dry Creek has indicated they are pleased with ODM2 Admin and plan to continue using it.

The CZIMEA project data management team consists of skilled developers, who developed their own scripts using python to load their ODM2 database using the ODM2PythonAPI https://github.com/ODM2/ODM2PythonAPI and the python library SQLAlchemy. They also contributed to the project, developing much of the mapping interface and linked their metadata with International Geo Sample Numbers (IGSNs) and ORCiDs. They have loaded over 2,000 soil sample specimens into their ODM2 instance. This project will complete loading soil specimen data into ODM2 in the coming year.

The Catalina-Jemez CZO (CJCGO) is set up similarly to the Luquillo CZO, has a skilled data manager Matej Durcik, experienced with database management. Matej used the ODM2 Admin web forms to input the metadata for the eight sites he has set up in ODM2 Admin so far. He also used the web forms to ingest data for the sites including 7 years of 10 minute data at most of the sites and 10 years of 30 minute data at some of the other sites. The CJCGO has indicated they are pleased with ODM2 Admin and would like to continue using it.

The Au Sable and Manistee watersheds have an extensive network of stream gages managed by Michigan State University. 38 pressure and temperature gages were loaded into an instance of ODM2, from flat files with a python script, in a process similar to that used with the Dry Creek experimental forest. Five years of 30 minute data were loaded for 23 sites. Several months into the ODM2 Admin pilot project, the data manager at Michigan State left for another job. We are in the process of assessing how to move forward with this site.

Findings

Through the course of the pilot project we interacted with researchers and data managers with varying skills in database administration, python, SQL and the other technologies used in the ODM2 Admin software stack. We also dealt with sites with varying degrees of study design complexity and different data management requirements. Two of the sites required little assistance in using ODM2 Admin, with experienced highly technical staff ODM2 can be deployed with ODM2 Admin providing essential tools to get started. The other two sites required additional assistance but that assistance was within the scope of the pilot and should be expected for many sites.

As seen in the case of the CJCGO ODM2 Admin can provide all of the functionality one needs to manage some sites. In the case of CZIMEA the tools provided by ODM2 Admin can function in more of a support role, allowing a user to validate data they have entered into ODM2 and
correct minor problems without writing scripts or complex SQL. When larger scale systematic changes are needed, scripts can be written using ODM2 Admin and the Django ORM or the ODM2PythonAPI. These scripting techniques can also be used to load ODM2 with complex metadata and data if desired.

Well documented long term studies like those managed by Michigan State and Boise State can also benefit from ODM2 Admin. Metadata required by ODM2 were gathered and ingested into ODM2 in a rather straight forward process. Michigan State has a large number of nearly identical sites, once one site is properly documented and it’s metadata entered into ODM2, it can be easily duplicated using ‘Save as New’ web form functions. The same thing can be done in scripts possibly faster but in practice, given the experience of the CJCZO and LCZO, not much faster unless done for 10s of new sites or time series. The experience with Dry Creek was similar to Michigan State, they have a wider variety of collected data but they follow a pattern familiar to a Critical Zone with stream gages, weather stations and soil sensors. Given the good documentation present these sites were readily loaded into ODM2.

Dry Creek had additional requirements, such as ingesting some data from data logger files while excluding others. Some tweaks to the user interface and command line interface tools were developed to accommodate these requirements. This has resulted in more robust data ingestion tools within ODM2 Admin, while these tools are implemented through the ODM2 Admin interface they can also be used on the command line or in scripts. In some cases the requests from Dry Creek may have seemed idiosyncratic but they overall provided a good test of the ODM2 Admin data loading functionality and resulted in data loading commands becoming more generalizable.

The experience of the ODM2 Admin pilot shows that while established sites with experienced staff can operate ODM2 using ODM2 Admin, new sites or projects, or those with junior staff, or minimal staff, will likely require assistance in getting started. Consultation with a data manager / developer experienced with ODM2 and ODM2 Admin are recommended for new projects. The amount of consultation required will depend on the complexity of the data to be managed but guidelines for a typical installation can be established.

Choosing to use ODM2 Admin provides sites with a number of advantages. Because of the highly flexible and well considered nature of the ODM2 information model, data streams of new discrete earth observations not planned at the outset of a project, can easily be incorporated. For new installations of ODM2 Admin on the CUAHSI cloud, users do not need to worry about server hardware, software updates, or data backups as these functions are handled automatically with the cloud host.

The information model and ODM2 Admin were developed with data management best practices in mind. Tools for data QA/QC, tracking changes in experimental setup through equipment changes, site visits, and maintenance, and publishing data as datasets with citations then
publishing to Hydroshare, all make managing data easier and can capably provide thorough metadata to fully explain observations and how they were generated.

As mentioned previously, ODM2 Admin and the ODM2 software ecosystem share a common software stack. These technologies are consistent with CUAHSI systems such as Hydroshare, which also makes use of ODM2. Additionally, ODM2 as an extension of ODM1 builds on the tools and accumulated knowledge developed for ODM1. Concepts and tools such as Water OneFlow web services can be deployed against ODM2 databases. Using tools like github, these common technologies affords the opportunity to build an open-source community of software development around ODM2 if CUAHSI would curate and maintain the software repository. This approach could leverage CUAHSI’s software development resources.

Although ODM2Admin uses a similar development environment as Hydroshare, some additional capacity building will be necessary to fully familiarize the CUAHSI staff with supporting ODM2Admin. This is not an immediate concern so long as Miguel remains affiliated with the project, but must be considered in the long run.

Recommendations

Given the results of the pilot project, we believe that ODM2 Admin is a viable data management system that will meet the needs of many academic researchers and should be eventually added as a CUAHSI data service.

The costs of this service include between $1500 to $2000 per year for cloud computing services, approximately 2 weeks for initial set-up (roughly $6000) and one week per year for the life of the project for consulting (roughly $3000). Labor is quoted as fully loaded (with benefits and overhead). A potential business model would be for projects to build in data management costs of roughly $10,000 into the first year of their project and then an additional $5,000 per year for the remaining years of the project that would be considered a consulting expense to CUAHSI. This is a small cost in the context of a $300,000 project and should provide good value-for-money to academics who don’t have data management systems. ODM2 Admin can also be made available to ongoing research sites or projects in a similar manner, although we anticipate new sites may achieve better results if they integrate ODM2 Admin into their initial work flow for site setup and documentation of observations and their metadata.

In the near term, we recommend that the ODM2 Admin pilot be continued for another year at a lower cost ($25,000 rather than $50,000) to enable Miguel to work with additional 3 or 4 sites to further refine cost estimates. The CUAHSI Biennial should be used to market these services to additional projects and highlight user experiences with the system to generate interest. Another decision to be made in the short term is whether to include these services into the renewal proposal (likely due in early 2018, before the biennial meeting is held).
We request the endorsement of the Informatics Standing Committee of these recommendations.
Appendix. ODM2Admin Documentation from GitHub Repository
ODM2 Admin

ODM2 Admin is an application for site level data management of environmental observations using Observation Data Model 2 (ODM2). The application was designed for management of data from the Luquillo Critical Zone Observatory located in northeastern Puerto Rico. For more details about why ODM2 Admin was developed see [Why ODM2 Admin was developed](https://github.com/ODM2/ODM2-Admin).

ODM2 was created through National Science Foundation Grant EAR-1224638. Support for the development of this application comes from NSF Grant EAR-1331841 Luquillo CZO.

The ODM2 Admin source code can be found here: [https://github.com/ODM2/ODM2-Admin](https://github.com/ODM2/ODM2-Admin)

ODM2 can be found here: [https://github.com/ODM2](https://github.com/ODM2)

Other ODM2 tools can be used in conjunction with ODM2 Admin, extensive testing has been done using ODM2 Admin with ODM2PythonAPI and WOFpy.

Django ORM models exist for all ODM2 tables. Web forms for ODM2Core and a number of additional ODM2 tables. Sites and sampling features are mapped via a leaflet mapping interface. Graphing of time series result values via highcharts are implemented. Data logger files can be imported with properly configured data logger file columns and time series results results.

Diagram of ODM2 Admin infrastructure:

![Diagram of ODM2 Admin infrastructure](image)

ODM2 Admin exists using a postgresql version of ODM2 data model, additional modifications may be needed to make this work with MSSQL or another database.

Extended infrastructure with WOFpy web services and ODM2PythonAPI
ODM2 Admin can be setup in conjunction with WOFpy and ODM2PythonAPI. WOFpy implements CUAHSI’s water one flow web services. See the [WOFpy github page](https://github.com/ODM2/WOFpy) for more. ODM2PythonAPI is a Python-based application programmer’s interface for the Observations Data Model 2 (ODM2). For more see the [ODM2PythonAPI github page](https://github.com/ODM2/ODM2PythonAPI).

An example postgresql database named ODM2AdminExamplePostgresqlDB is provided, this is a custom postgresql format backup which can be restored to an empty database. An extrasql.sql file contains some extra views used for efficiently exporting data as emails.

**Primary Installation**

The fastest way to get started is to install with DockerHub:

- [ODM2 Admin Docker Image Creation](https://github.com/ODM2/ODM2-Admin)

Alternatively you can download the source code from github ([https://github.com/ODM2/ODM2-Admin](https://github.com/ODM2/ODM2-Admin)) setup a conda environment, create an ODM2 database, run the extrasql.sql script (found in the root directory of the source code) on that database, and change settings in:

```
ODM2-Admin-master\templatesAndSettings\settings\base.py
ODM2-Admin-master\templatesAndSettings\settings\development.py
```

When you are deploying to production you will want to change the settings in production.py instead of development.py. You can also have the settings files point to an existing ODM2 database, you will need to run the extrasql.sql on the database. For more details on settings see [ODM2 Admin Settings](https://github.com/ODM2/ODM2-Admin).
```
pip install -r requirements.txt
```

or
```
conda config --add channels conda-forge --force
conda create -n ENVNAME python=2.7 --file requirements.txt
```

or create the conda environment with the development requirements as well:
```
conda create -n ENVNAME python=2.7 --file requirements.txt --file requirements-dev.txt --channel conda
```

You will need to run `extrasql.sql` on a PostgreSQL instance of ODM, a blank schema script can be found here: https://github.com/ODM2/ODM2/tree/master/src/blank_schema_scripts/postgresql

Six instances of ODM2 Admin have been deployed

- http://odm2admin.ohsu.org/LCZO/mapdata.html
- http://odm2admin.ohsu.org/DryCreek/mapdata.html
- http://odm2admin.ohsu.org/TRACE/TRACE/mapdata.html
- https://dev-odm2admin.ohsu.org/CZIMEA/mapdata.html

Another deployment for demonstration purposes is available, the test ODM2 Admin Walkthrough for more information about this:

https://dev-odm2admin.ohsu.org/Sandbox/

**Using ODM2 Admin**

The documents below provide instructions on using ODM2 Admin.

- **Motivation for ODM2 Admin**
- **Getting Started With ODM2 Admin**
- **configuring ODM2 Admin Settings**

**ODM2 Admin Walkthrough**

The ODM2 Admin Walkthrough uses a Sandbox instance of ODM2 Admin while also documenting the work through steps with detailed descriptions and images from the sandbox. Many of the features of ODM2 Admin are described here.

- **ODM2 Admin Walkthrough**
- **ODM2 Admin Forms**
- **Managing Profile Results With ODM2 Admin**
- **Time series QA/QC**
- **Data Sharing and Visualization Tips**
- **ManagingODM2WithTheDjangoORM**

**Docs Home**

- **ODM2 Admin docs home page**
- **Search the docs**

https://odm2.github.io/ODM2-Admin/
ODM2 Admin Walkthrough

Below is a tutorial which explains many of the features of ODM2 Admin. This tutorial was developed for the 2017 BiG-CZ / ODM2 Hands-On Workshop, November 15-16, 2017 at UC Riverside, CA. It is meant as a demonstration of the capabilities of ODM2 Admin and as a use case for ODM2. Links contained here access a demonstration version of ODM2 Admin populated with data from the Luquillo Critical Zone Observatory and hosted by CUAHSI. If you would like access to the sandbox please email leonmi@sas.upenn.edu.

You can login to the sandbox here: https://dev-odm2admin.cuahs.org/Sandbox/

After logging in you will see the ODM2 Admin Home page:

This page consists of several parts, links to log out and change your password, the ODM2 Admin Shortcuts, ODM2 Admin administration, authentication and authorization, and recent actions.

ODM2 Admin Shortcuts

Upon logging into ODM2 Admin you will see the below navigation shortcuts across the top.

- The first shortcut, displayed here as ‘ODM2 Admin’ with a cog icon, provides a list of all the ODM2 pages.
- The second, ‘Add Sensor Data’, provides links to where you should enter information if you are trying to add new sensor data or you want to make changes to information relevant to sensor data.
- Third, ‘Add Soil Profile Data’ provides links for adding or editing Profile result data.
- Fourth, ‘Record an Action’, provides links for adding or editing actions and methods (standardized method for how to perform an action).
- Fifth, Manage Citations, provides links for managing and exporting citations.
- Sixth, Graph My Data, provides links for data plotting and a map of your sites.
The ODM2 Admin Administration link will take you to a list of all 44 ODM2 entities that can be directly managed in ODM2 Admin. See the ODM2 Admin Forms for details about the ODM2 Admin Forms [ODM2 Admin Forms](http://odm2.github.io/ODM2-Admin/ODM2AdminDemo.html).

Recent actions show changes you have recently completed in ODM2 Admin such as adding or editing an item in the forms.

Additional entities exist within the ODM2 information model these need to be managed with another tool such as with the ODM2PythonAPI. Django models exist for each ODM2 entity so it is also possible to write Python scripts using the Django Object relational mapper. See [The Django ORM and WOPpy](http://odm2.github.io/ODM2-Admin/ODM2AdminDemo.html) for details on using the Django ORM with ODM2 databases.

- 1) The ODM2 Admin Forms
- 2) Using Data Logger Files
- 3) Managing Profile Results With ODM2 Admin
- 4) Time series QA/QC
- 5) Data visualization and URL parameters
- 6) Using the Django ORM
- ODM2 Admin docs home page
- Search the docs
Data Logger Files

Ingesting data by uploading and processing data logger files is a key feature of ODM2 Admin. While logged into the ODM2 Sandbox you can navigate to https://dev-odm2admin.cuahsi.org/Sandbox/odm2admin/dataloggerfiles/

1. Select the first data logger file ending in QP.DO https://dev-odm2admin.cuahsi.org/Sandbox/odm2admin/dataloggerfiles/133/change/.

2. Download and open the file, we can see that the file has 3 columns, Date Time GMT 04:00, DO Conc, and Temp the data logger file model correspondingly also has 3 data logger file columns.

3. Click SHOW on the first data logger file column, this data logger file column has a data result, an instrument output variable and a Column label. The column label must match a column label in the file for the data logger file to be ingested. The data logger file column also has a recording interval and a recording interval units as well as an aggregation statistic.

4. The Data result contains much of the information about how our time series is configured. Go to the data results page (https://dev-odm2admin.cuahsi.org/Sandbox/odm2admin/results/) to find our result. If we search for Prieta, the name of the site, we will find our data result as in the below image.
5. Now select the data result, with the variable DO Concentration and ID 21 (https://dev.odm2admin.cuahsi.org/Sandbox/odm2admin/results/21/change/).

6. We can see that here the sampling feature / location action is defined, the Result type (time series coverage), The variable, units, processing level, as well as other fields many of them optional. We also see Time Series Results, Measurement Results and Profile Results. Because this data result defines a time series it needs to have an associated time series result.

7. We can click the + icon to add a variable, unit processing level or other field, we can click the pencil icon to edit the current item.

The Sensor dashboard:

We have a dashboard view of sensors which can also be configured in the settings files.

Here is an example dashboard for DryCreek Experimental Watershed.

http://odm2admin.cuahsi.org/DryCreek/sensordashboard/featureaction=20/
ODM2 Admin Managing Profile Results

Some of the sampling features, accessible from the ODM2 Admin Map contain sampling features such as

Field area Palm SJH, WS 2015 represents a set of soil pits making up a study area. Soils taken from this study area were analyzed for Acid phosphatase, Ammonium oxalate extractable aluminum and other compounds. A graph of these values can be viewed here: https://dev-odm2admin.cuahsi.org/Sandbox/profilegraph/selectedrelatedfeature=15/popup=true/. Follow these steps:

# Select Variable Codes: Clay, Sand, and Silt
# Click 'Update Chart' wait for the page to refresh
# click 'Export Data'.

We can export the data from this page then we get the below, the method here has been shortened for space.

Palm SJH, WS 2015 Clay values

<table>
<thead>
<tr>
<th>databaseid</th>
<th>depth</th>
<th>sampling feature/location</th>
<th>sampling feature uri</th>
<th>method</th>
<th>citation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

http://odm2.github.io/ODM2-Admin/ProfileResults.html#profileresults
<table>
<thead>
<tr>
<th>databaseid</th>
<th>depth</th>
<th>sampling feature/location</th>
<th>sampling feature uri</th>
<th>method</th>
<th>citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>6196</td>
<td>0.0-10.0 CM-Centimeter</td>
<td>Elfin-Plot 61-Block 5-SJH-WS-2015</td>
<td></td>
<td>We tested the influence of anaerobiosis...</td>
<td>13.40865635</td>
</tr>
<tr>
<td>6198</td>
<td>10.0-20.0 CM-Centimeter</td>
<td>Elfin-Plot 61-Block 5-SJH-WS-2015</td>
<td></td>
<td>We tested the influence of anaerobiosis...</td>
<td>14.3594963551</td>
</tr>
<tr>
<td>6200</td>
<td>0.0-10.0 CM-Centimeter</td>
<td>Elfin-Plot 62-Block 5-SJH-WS-2015</td>
<td></td>
<td>We tested the influence of anaerobiosis...</td>
<td>12.8595966657</td>
</tr>
</tbody>
</table>

Palm SJH, WS 2015 Sand and Silt values

<table>
<thead>
<tr>
<th>‘Sand ·unit-Percent-processing level-L1 passed QAQC’</th>
<th>‘Silt ·unit-Percent-processing level-L1 passed QAQC ·L1 passed QAQC’</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.43426367</td>
<td>44.5807837</td>
</tr>
<tr>
<td>9.447758926</td>
<td>55.08999424</td>
</tr>
<tr>
<td>2.758360072</td>
<td>48.96050472</td>
</tr>
</tbody>
</table>

The export contains a database id record for the profile result value, the depth which is the difference between the current profile result value's intended depth and the previous profile result value intended depth where the profile results have the same sampling feature.

Let’s look at another data export for samplingfeature PALMDYS-21, If we go to the following link:

https://dev-odm2admin.cuahsi.org/Sandbox/profilegraph/selectedrelatedfeature=674/popup=true/

PALMDYS-21 is a field area where 9 soil pits were dug. Again let’s Select Variable Codes: Clay, Sand, and Silt, Click ‘Update Chart' wait for the page to refresh and click ‘Export Data'.
This export also contains sampling feature URIs, in this case these are International Geo Sample Numbers (IGSN) issued by the Interdisciplinary Earth Data Alliance (IEDA) [http://www.geosamples.org/](http://www.geosamples.org/)

### PALMDYS-21 locations with IGSNs

<table>
<thead>
<tr>
<th>'sampling feature/location'</th>
<th>'sampling feature uri'</th>
</tr>
</thead>
<tbody>
<tr>
<td>'PALMDYS-21-P118 Ridge'</td>
<td><a href="http://app.geosamples.org/sample/igsn/IELCZP118">http://app.geosamples.org/sample/igsn/IELCZP118</a></td>
</tr>
<tr>
<td>'PALMDYS-21-P120 Valley'</td>
<td><a href="http://app.geosamples.org/sample/igsn/IELCZP220">http://app.geosamples.org/sample/igsn/IELCZP220</a></td>
</tr>
<tr>
<td>'PALMDYS-21-P121 Ridge'</td>
<td><a href="http://app.geosamples.org/sample/igsn/IELCZP221">http://app.geosamples.org/sample/igsn/IELCZP221</a></td>
</tr>
<tr>
<td>'PALMDYS-21-P122 Slope'</td>
<td><a href="http://app.geosamples.org/sample/igsn/IELCZP222">http://app.geosamples.org/sample/igsn/IELCZP222</a></td>
</tr>
<tr>
<td>'PALMDYS-21-P123 Valley'</td>
<td><a href="http://app.geosamples.org/sample/igsn/IELCZP223">http://app.geosamples.org/sample/igsn/IELCZP223</a></td>
</tr>
<tr>
<td>'PALMDYS-21-P125 Slope'</td>
<td><a href="http://app.geosamples.org/sample/igsn/IELCZP225">http://app.geosamples.org/sample/igsn/IELCZP225</a></td>
</tr>
<tr>
<td>'PALMDYS-21-P126 Valley'</td>
<td><a href="http://app.geosamples.org/sample/igsn/IELCZP226">http://app.geosamples.org/sample/igsn/IELCZP226</a></td>
</tr>
</tbody>
</table>

### 4) Time series QA/QC

- [ODM2 Admin docs home page](http://app.geosamples.org)
- Search the docs

---

[http://edm2.github.io/ODM2-Admin/ProfileResults.html#profileresults](http://edm2.github.io/ODM2-Admin/ProfileResults.html#profileresults)