open-data-etl-tool-kit Documentation

Release 1.0.0

City of Chicago

December 09, 2015

Contents

CHAPTER 1

Installation & Configuration

This section outlines basic installation procedures for Kettle, the open data ETL framework, and other necessary components. This section also discusses some suggested configurations which will allow for easier maintenance over time.

Installation consists of three parts

- Cloning or installing toolkit repository
- Installing Kettle (or Pentaho)
- Installing Socrata DataSync
- Configuring Kettle and DataSync installation

1.1 Installation on Mac OS X, Linux, or Unix

1.1.1 Installing ETL framework

First, determine a location of the installation of the toolkit. All scripts, programs, and transformations related to ETL processes will remain in this directory.

Using `git`:

```
$ cd /path/to/directory
$ git clone
```

Alternatively, one can download the zip file from GitHub and extract the contents to the above directory.

1.1.2 Installing & configuring Kettle

Next, we will need to obtain Kettle or Pentaho. Download Kettle to your computer.

The Kettle installation should allow for easy upgrades to the data integration software without needing to reconfigure any ETLs. Likewise, upgrading to new versions should permit testing. Therefore, Kettle should be installed to a version-specific folder, such as `data-integration-x.y.z`. Assuming the zipped file is located in `~/Downloads`, one could run the following:

```
$ cd ~/Downloads
$ unzip pdi-ce-4.4.0-stable.zip -d path/to/directory/open-data-etl-utility-kit
$ mv data-integration data-integration-4.4.0
```

Create a symlink between `data-integration` and the current version:

```
$ cd /path/to/directory/open-data-etl-utility-kit
$ ln -s data-integration-x.y.z data-integration
```

A future version of Kettle can be installed and tested in its own directory without impacting production. Once ETLs are ready to use a newer version, update the symlink to the appropriate directory.

1.1.3 Installing DataSync

This framework uses Socrata DataSync to post data to the portal. This utility is *only* compatible with Socrata portals. Fortunately, this utility handles incremental updates and upserting without additional logic in the ETL.

This framework will work with the version of DataSync in the DataSync directory of this repository, which is named in a way to make the version clear. It should be renamed to `datasync.jar` in an actual deployment.

You may install DataSync to any directory. Later configuration will direct Kettle to the correct location.

You can configure DataSync to run on a "headless" Linux machine–a Linux server which is only accessible through a command prompt. Running DataSync on a headless machine requires configuration to pass the domain, username, password, and token without a graphical user interface (GUI). Instructions on configuring a headless is available on the DataSync support site. The `DataSync` directory contains templates for the config.json file

If you installed DataSync to another directory, such as `/path/to/DataSync`, then you must edit `/path/to/directory/open-data-etl-utility-kit/DataSync/load_preferences.sh`. Specifically, the script must now read:

```
java -jar /path/to/DataSync/datasync.jar --config config.json --jobType LoadPreferences
```

In order for later automation.

1.1.4 Setting-up default directories

Users will need to define two environmental variables for their Kettle installation:

```
* Location of the ETL directory (e.g., /path/to/directory/open-data-utility-kit)
* Location of the DataSync installation (e.g., /path/to/directory/open-data-utility-kit/DataSync)
```

This configuration will only need to be adjusted once for each environment. It will also allow for each deployment of ETLs across multiple operating systems without needing to configure the ETL itself.

Launch Kettle by finding and launching spoon.sh, or, run the following in a command prompt:

```
> sh /path/to/directory/open-data-utility-kit/data-integration/spoon.sh
```

Once Kettle launches, selected Edit > Edit the kettle.properties file:

Right-click to insert a new line. Once a blank line is available, add `ETL_DIRECTORY` as a variable name and add the path to your ETL directory under value (e.g., `/path/to/directory/open-data-utility-kit`).

Add another line and enter `DATASYNC_DIRECTORY` as a variable name and `/path/to/directory/open-data-utility-kit/DataSync`

The `kettle.properties` file can also be manually edited. It is typically located under the following directories, depending on your current version of Windows:

```
$HOME\.Kettle
```

Navigate to the appropriate location and open `kettle.properties`. Add the following lines to the file and save:

```
ETL_DIRECTORY = /path/to/DataSync
DATA_SYNC_DIRECTORY = /path/to/directory/open-data-utility-kit/DataSync
```

1.2 Installation on Windows

1.2.1 Installing ETL framework

First, determine a location of the installation of the toolkit. All scripts, programs, and transformations related to ETL processes will remain in this directory.

Using `git`:

```
$ cd C:\path\to\directory
$ git clone
```

Alternatively, one can download the zip file from GitHub and extract the contents to the above directory.

1.2.2 Installing & configuring Kettle

Next, we will need to obtain Kettle or Pentaho. Download Kettle to your computer.

The Kettle installation should allow for easy upgrades to the data integration software without needing to reconfigure any ETLs. Likewise, upgrading to new versions should permit testing. Therefore, Kettle should be installed to a version-specific folder, such as `data-integration-x.y.z`.

Install Kettle to `data-integration-x.y.z`, where `x.y.z` is the version number (e.g., 4.4.0).

Create a link between `data-integration` and the current version:

```
> cd C:\path\to\directory\open-data-etl-utility-kit
> mklink /j "data-integration-x.y.z" "data-integration"
```

A future version of Kettle can be installed and tested in its own directory without impacting production. Once ETLs are ready to use a newer version, update the symlink to the appropriate directory.

1.2.3 Installing DataSync

This framework uses Socrata DataSync to post data to the portal. This utility is *only* compatible with Socrata portals. Fortunately, this utility handles incremental updates and upserting without additional logic in the ETL.

1.2.4 Setting-up default directories

Users will need to define two environmental variables for their Kettle installation:

- Location of the ETL directory (e.g., C:pathtodirectoryopen-data-etl-utility-kit)
- Location of the DataSync installation (e.g., C:pathtodirectoryopen-data-etl-utility-kitDataSync) This configuration will only need to be adjusted once for each environment. It will also allow for each deployment of ETLs across multiple operating systems without needing to configure the ETL itself.

Launch Kettle by finding and launching spoon.bat, or, run the following in a command prompt:

```
> C:\path\to\directory\open-data-etl-utility-kit\data-integration\spoon.bat
```

Once Kettle launches, selected Edit > Edit the kettle.properties file:

Right-click to insert a new line. Once a blank line is available, add `ETL_DIRECTORY` as a variable name and add the path to your ETL directory under value (e.g., `C:/path/to/directory/open-data-utility-kit`).

Add another line and enter `DATA_SYNC_DIRECTORY` as a variable name and `C:/path/to/directory/open-data-utility-kit/DataSync`. It is recommended to use forward-slashes to maintain compatibility with Linux deployment.

The `kettle.properties` file can also be manually edited. It is typically located under the following directories, depending on your current version of Windows:

```
| *Windows:* C:\Documents and Settings\<username>\.kettle\
| *Windows Vista and after:* C:\Users\<username>\.kettle
```

Navigate to the appropriate location and open `kettle.properties`. Add the following lines to the file and save:

```
ETL_DIRECTORY = C:/path/to/directory/open-data-etl-utility-kit
DATA_SYNC_DIRECTORY = C:/path/to/directory/open-data-etl-utility-kit/DataSync
```

1.3 Understanding repository layout

After completing this section, the framework should resemble the following structure. Several scripts use relative directories dependant on the following layout. Any deviation will require some, but simple, reconfiguration.

- open-data-etl-kit
  - ETL
    - Utilities
  - Log
  - DataSync
  - Tools
  - data-integration
  - data-integration-x.y.z

`open-data-etl-kit` may be renamed to meet your preferences. Each directory will have the following responsibilities:

- **ETL** - will contain subfolders pertaining to each ETL (e.g., hello-world). These directories will contain the logic necessary to extract and transform the data for the portal. If you use our templates, each ETL will call to the `Utilities` directory to complete additional tasks.
- **ETL/Utilities** - will contain generic steps used by ETLs, such as sending email alerts and preparing OS-level variables to use with DataSync.
- **Log** - The recommended setup will direct Kettle log files to this directory using the ETL name and timestamp. If desired, it can serve as a historical repository of ETL performance and logs for diagnostics. This directory contains serveral bash scripts (Linux/MacOS X/Unix only) that make it easier to find or evaluate the logs for specific ETLs.
- **DataSync** - contains configuration files for DataSync. The actual DataSync installation can be placed in your preferred directory.
- **Tools** - contains tools to help with administering ETL processes.
- **data-integration** - a link which directs to the directory of Kettle being used.

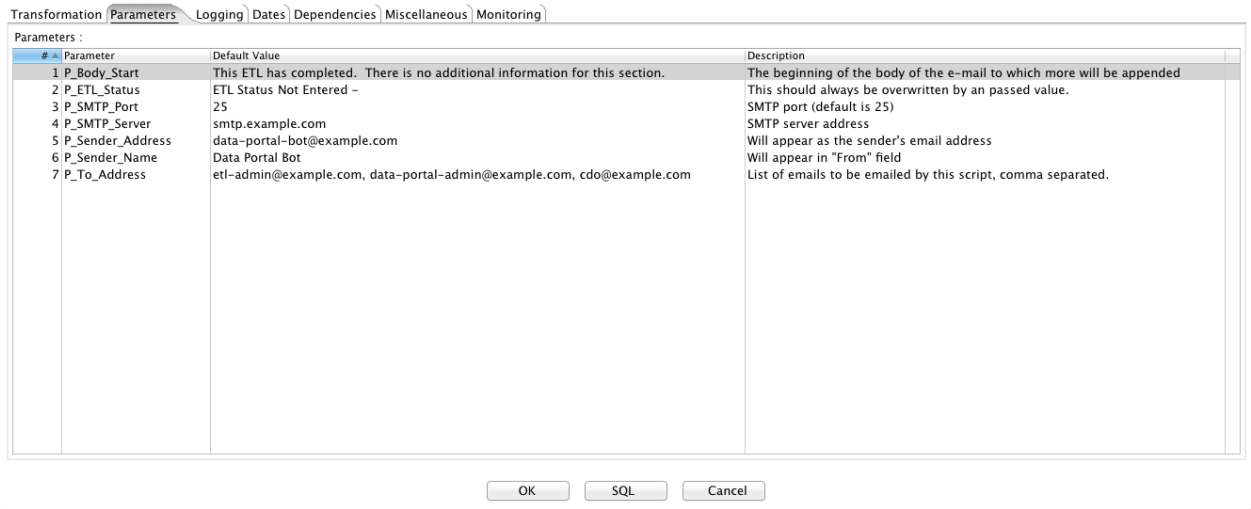
- **data-integration-x.y.z** - the Kettle application files.

1.4 Setting-up Email

Open `open-data-etl-kit/ETL/Utilities/ETL_Completion_E-Mail.ktr` in Kettle. Select `Edit -> Settings` and select the Parameters tab. Enter the appropriate values for:

- **P_SMTP_Port** - SMTP port (default is 25)
- **P_SMTP_Server** - SMTP server address. The machine running the ETL will need be able to access that server
- **P_Sender_Address** - Will appear as the sender's email address
- **P_Sender_Name** - Will in the "From" field.
- **P_To_Address** - List of emails, comma separated.

The **P_Body_Start** and **P_ETL_Status** parameters contain default values for the e-mail message. You may edit them if you wish but it is not important to do so because they should always be overwritten with real values when the ETL runs.



CHAPTER 2

Creating & Configuring an ETL

This section will walk-through the creation of an ETL. Launching a new ETL requires the following steps:

- Prepare the end-point (e.g., Socrata) with the columns and data set name.
- Configure the ETL parameters (e.g., 4x4 ID) and control file.
- Write an ETL

2.1 Initial preparation

First, the end-point must be configured. With a Socrata portal, create all of the columns with API field names. If you have an initial extract, a recommended workflow is to upload it to the dataset. Otherwise, the columns can be created manually by the user.

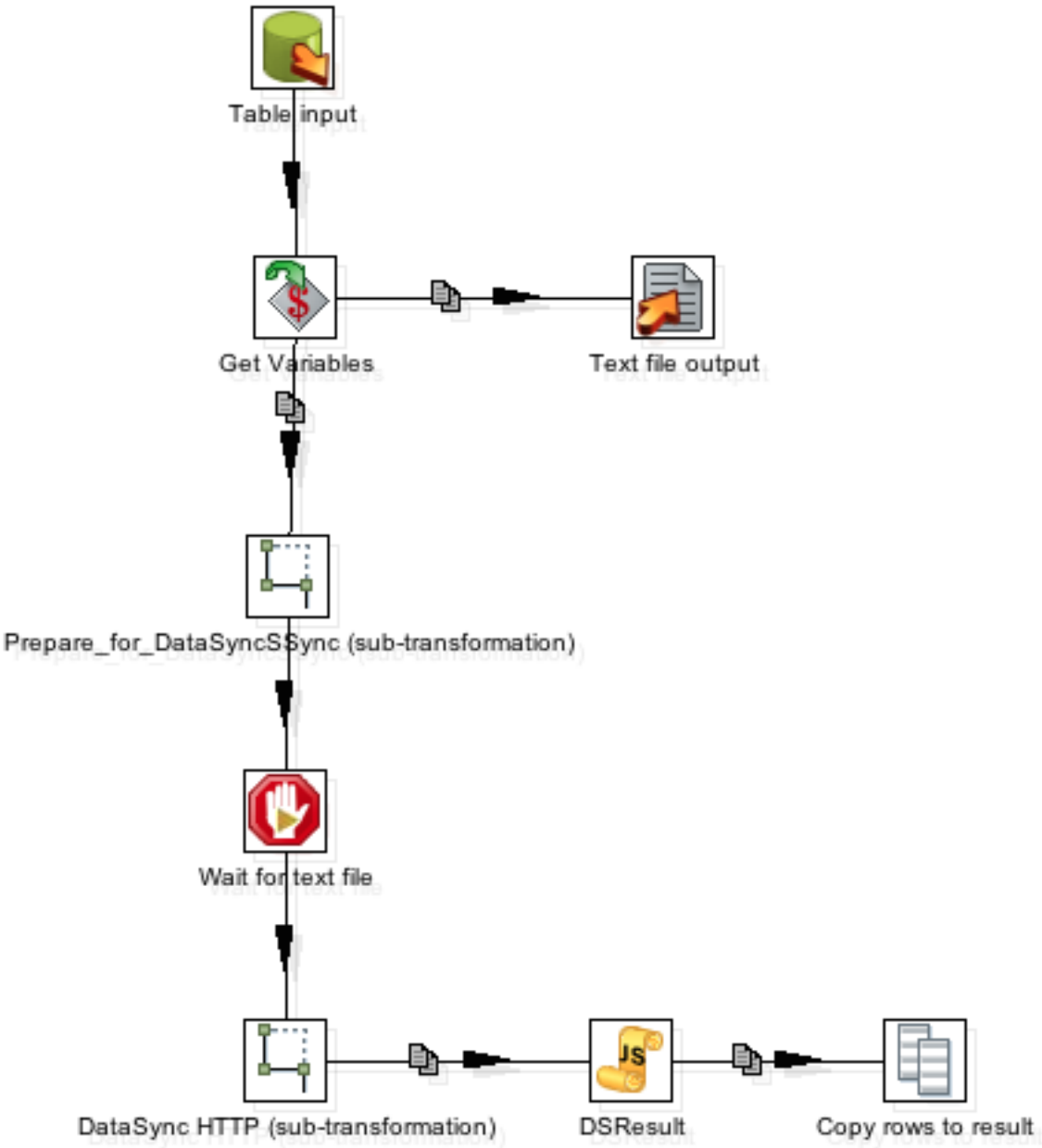
Find the Socrata 4x4 of the newly created dataset.

Create a new folder with the dataset name and 4x4 in the `ETL` directory (i.e., `open-data-etl-utility-kit/ETL/Data_Set_Name_abcd-1234`).

2.2 Write ETL

A basic template of a new ETL is at `open-data-etl-utility-kit/ETL_Template.ktr`. Copy it to the new directory with a name maching the directory, such as:

```
$ cp ETL_Template.ktr ETL/Data_Set_Name_abcd-1234/Data_Set_Name_abcd-1234.ktr
```



Open the file in Kettle. Several steps are included, but the following items should not be modified:

- Get Variables
- Prepare_for_DataSyncSSync (sub-transformation)
- Wait for text file
- DataSync HTTP (sub-transformation)
- DSResult
- Copy rows to result
- Text file output

Users should modify the data extraction (such as Read Table or JSON Input) and any custom transformation.

2.3 Configuring ETL parameters

In Kettle, select `Edit -> Settings...`, then click on the "Parameters" tab. Fill-in the appropriate fields for each parameter, typically along the following lines:

- **P_ControlFile** - Name of control file (e.g., Data_Set_Name_control.json)
- **P_DatasetID** - Dataset 4x4 (e.g., abcd-1234)
- **P_File** - File name of the file DataSync should use for the update (e.g., Data_Set_Name_abcd-1234.csv)

Move the DataSync configuration template file (`_control.json`) to the new directory, renaming it in the process:

```
$ cp _config.json ETL/Data_Set_Name_abcd-1234/Data_Set_Name_control.json
```

Edit the configuration file by inserting the appropriate API field names.

2.4 Suggested naming conventions

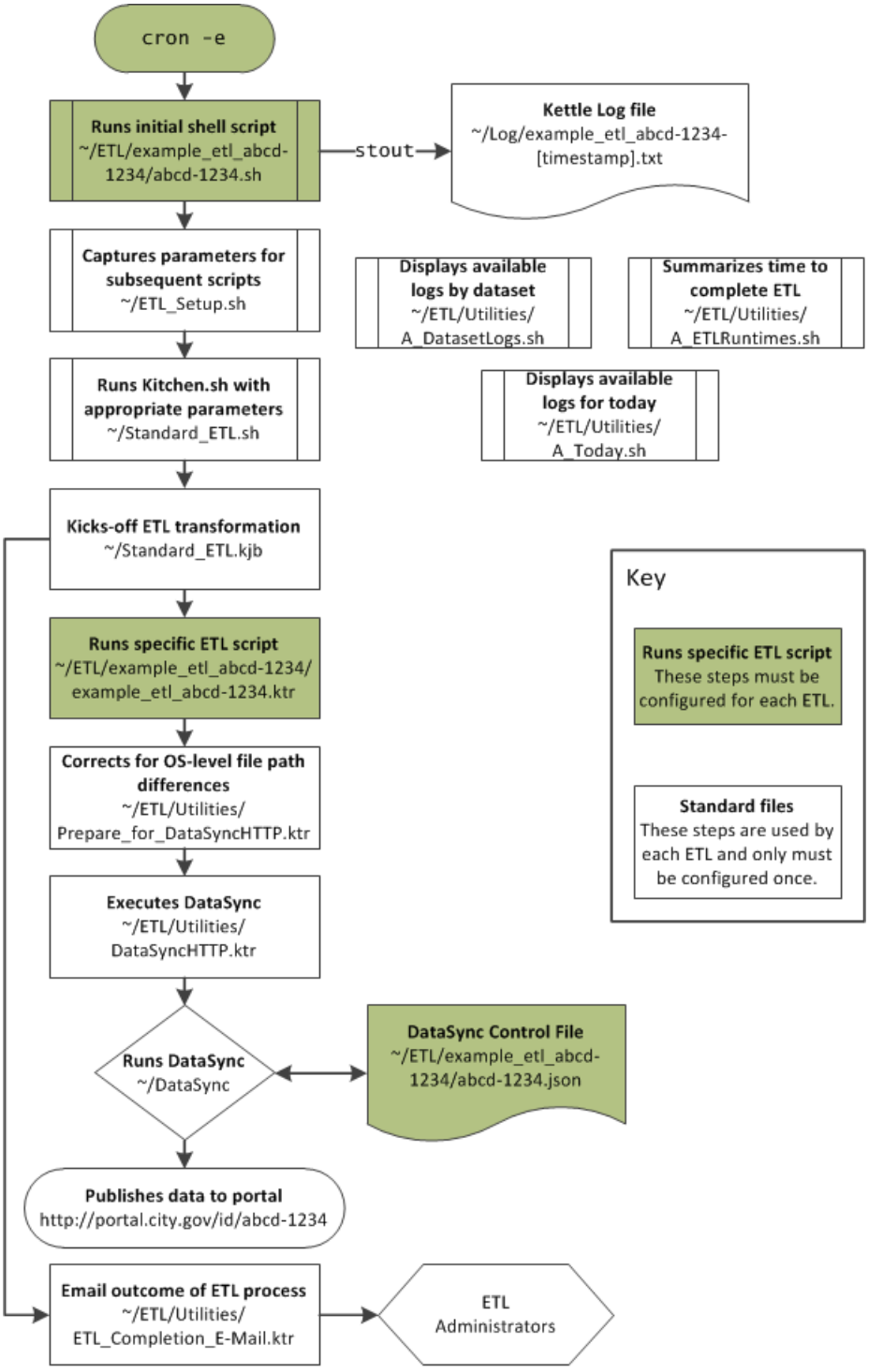
It can be difficult to manage dozens, if not hundreds, of ETLs. The City of Chicago data science team names each folder and Kettle transformation file with the same naming schema: `Name_of_file_abcd-1234`, where `abcd-1234` is the unique four-by-four of the dataset. For instance, the city's crime data is saved under the folder `Crimes_2001_to_present-ijzp-q8t2`.

CHAPTER 3

Setting-up Automation

This section describes how to setup automation using Kettle. This is currently dependent on a number of bash scripts and is only compatible with MacOS X, Linux, and Unix operating systems.

There are a number of files and transformations used to support the entire ETL. The diagram below shows the workflow and relationship between various files.



The automated process is initiated using a bash script, `Setup-Script.sh`, to kick-off scripts. The author also needs to setup the timing of the scripts using cron jobs.

3.1 Configurating Setup-Script.sh (e.g., abcd-1234.sh)

`Setup-Script.sh` is a standard template to be copied and used with each ETL. Suppose the dataset as a four-by-four of "abcd-1234", then begin by moving the template to the appropriate directory:

```
$ cd /path/to/directory/open-data-etl-utility-kit
$ cp Setup-Script.sh Data_Set_Name_abcd-1234/abcd-1234.sh
```

Each file will need to be edited with the appropriate the name of the KTR file without the extension `ETL_Name` and the directory containing the ETL. Using the above example, the file should be edited with the following lines:

```
ETL_NAME=ata_Set_Name_abcd-1234
ETL_DIR_RELATIVE=Data_Set_Name_abcd-1234/
```

After editing, save the file.

Finally, edit `open-data-etl-utility-kit/ETL/ETL_Setup.sh` to include the file path to Java and the ETL directory:

```
PATH=$PATH:/path/to/jdk/bin
ETL_ROOT_DIR=/path/to/directory/open-data-etl-utility-kit/ETL
```

3.2 Setting-up Timing

The timing of the automated script is manged through cron jobs. Edit the cron job manager in the terminal by typing `crontab -e` in the shell. The cron job contains the starting script and also instructs the logging to be directed to the appropriate log files. For example:

```
* * * * * /path/to/directory/open-data-etl-utility-kit/ETL/Data_Set_Name_abcd-1234/abcd-1234.sh >> /p
```

The astrisks should be edited to meet the desired update schedules. A quick guides of those settings can be found on Wikipedia.

3.3 Testing process

A simple way to test the process is to execute the following line in the command prompt:

```
/path/to/directory/open-data-etl-utility-kit/ETL/Data_Set_Name_abcd-1234/abcd-1234.sh >> /path/to/di
```

If correctly configured, the dataset should be updated, log files should be updated, and users should receive email alerts.

CHAPTER 4

Utilities for Administering ETLs

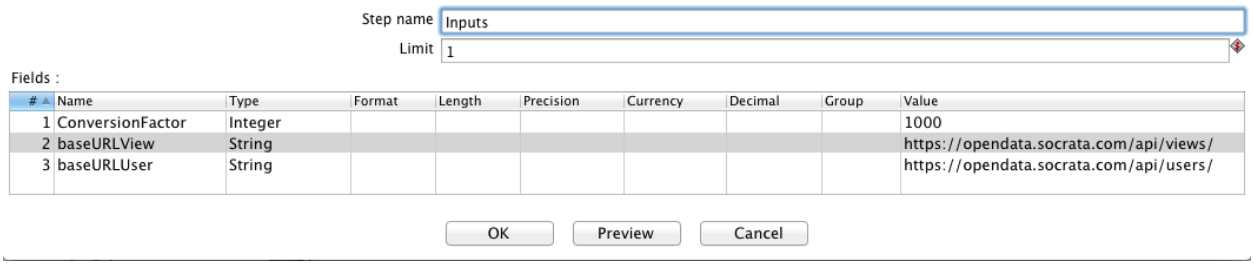
This repository contains several helpful tools to assist with monitoring ETL processes.

4.1 Checking last-updated information

File: Show_Update_Times_of_View.ktr

Description: A Kettle transformation to be run in Spoon (the Kettle GUI). It pulls update information about a dataset from Socrata APIs and presents it in a human-readable format.

Usage: Open Show_Update_Times_of_View.ktr in Kettle/Pentaho. For the initial setup, open the "Inputs" step and replace baseURL parameter with the URL for a specific data portal (e.g., opendata.socrata.com, data.cityofchicago.org).



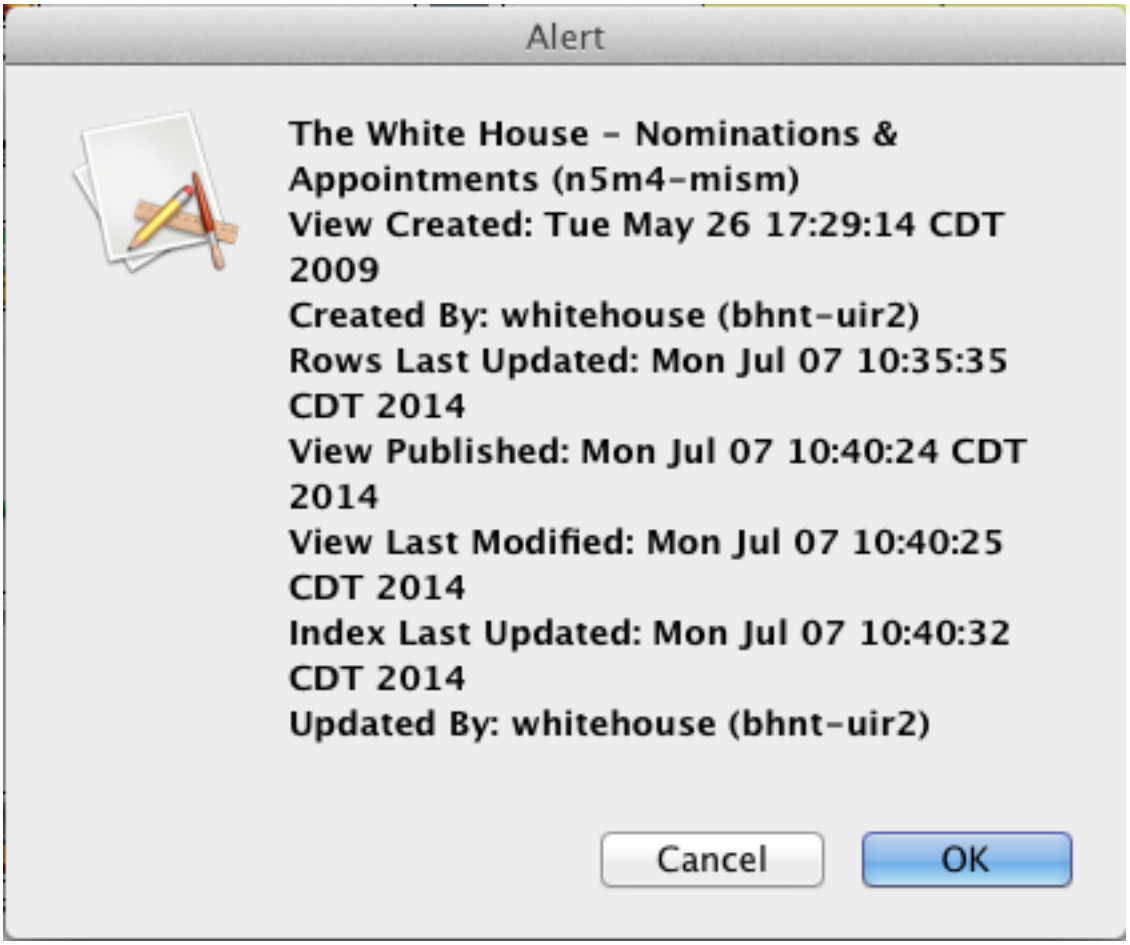
To execute, run the transformation (F9) and input the appropriate 4x4 as a value for the PARAM_four-by-four parameter, then press "OK"

Parameters

#	Parameter	Value	Default value	
1	PARAM_four-by-four	n5m4-mism		

Returns: If successful, it will return a prompt window with the following fields:

- Dataset/view title
- Creation date and time
- Dataset/view author
- Last-updated date and time
- Published date and time
- Last modified date and time
- Index last modified date and time
- Username who last updated data



4.2 Show all log files

File: Log/A_DatasetLogs.sh (MacOS X/Linux/Unix only)

Description: Shows all of the log files associated with a dataset.

Usage: Open the terminal and type the name of a dataset:

```
$ cd /path/to/directory/open-data-etl-utility-kit/
$ sh Log/A_DatasetLogs.sh Name_of_dataset
```

Returns: Will list the log files associated for a user-specified ETL job. The output is displayed in the terminal.

File: Log/A_DatasetLogs.bat (Windows only)

Description: Shows all of the log files associated with a dataset.

Usage: Open the command prompt window and type the name of a dataset:

```
> cd \path\to\directory\open-data-etl-utility-kit\
> \Log\A_DatasetLogs.bat Name_of_dataset
```

Returns: Will list the log files associated for a user-specified ETL job. The output is displayed in the command prompt window.

4.3 Summarize ETL run times

File: Log/A_ETLRuntimes.sh (MacOS X/Linux/Unix only)

Description: Shows the runtime for ETLs with a dataset.

Usage: Open the terminal and type the name of a dataset:

```
$ cd /path/to/directory/open-data-etl-utility-kit/
$ sh Log/A_ETLRuntimes.sh Name_of_dataset
```

Returns: The output will show the total run-times recorded in log files for the user-specified ETL. The output is displayed in the terminal.

File: Log/A_ETLRuntimes.bat (Windows only)

Description: Shows the runtime for ETLs with a dataset.

Usage: Open the command prompt window and type the name of a dataset:

```
> cd \path\to\directory\open-data-etl-utility-kit\
> Log\A_ETLRuntimes.bat Name_of_dataset
```

Returns: The output will show the total run-times recorded in log files for the user-specified ETL. The output is displayed in the command prompt window.

4.4 Show today's ETL logs

File: Log/A_TodayLogs.sh (MacOS X/Linux/Unix only)

Description: Shows log files which were created today

Usage: Open the terminal and run the command:

```
$ sh /path/to/directory/open-data-etl-utility-kit/Log/A_TodayLogs.sh [-e]
```

Returns: The output will show the list of log files which were generated today. With the *-e* parameter, a group of datasets specified in a parameter at the beginning of the script will be excluded (generally, those that run frequently and would clutter the output). The output is displayed in the terminal.

File: Log/A_TodayLogs.bat (Windows only)

Description: Shows log files which were created today

Usage: Open the command prompt window and run the command:

```
> sh \path\to\directory\open-data-etl-utility-kit\Log\A_TodayLogs.bat [-e]
```

Returns: The output will show the list of log files which were generated today. With the *-e* parameter, a group of datasets specified in a parameter at the beginning of the script will be excluded (generally, those that run frequently and would clutter the output). The output is displayed in the command prompt window.

4.5 Run a specific ETL

File: Log/A_RunETL.sh (MacOS X/Linux/Unix only)

Description: Performs a one-time run of an ETL normally run on a scheduled basis through the crontab file. This file need not be in the Log directory to run correctly. It does not use the log files and is in the Log directory only to keep it with other scripts.

Usage: Open the terminal and type the name of a dataset:

```
$ cd /path/to/directory/open-data-etl-utility-kit/
$ sh Log/A_RunETL.sh Name_of_dataset
```

Returns: The script will find and run the ETL command for the specified dataset. The output will show the command run so the user can confirm it was the intended dataset ETL.

File: Log/A_RunETL.bat (Windows only)

Description: Performs a one-time run of an ETL normally run on a scheduled basis through by the Windows task scheduler. This file need not be in the Log directory to run correctly. It does not use the log files and is in the Log directory only to keep it with other scripts.

Usage: Open the terminal and type the name of a dataset:

```
$ cd \path\to\directory\open-data-etl-utility-kit\
$ Log\A_RunETL.bat Name_of_dataset
```

Returns: The script will find and run the ETL command for the specified dataset. The output will show the command run so the user can confirm it was the intended dataset ETL.

This toolkit provides several utilities and framework to help governments deploy automated ETLs using the open-source Pentaho data integration (Kettle) software.

Namely, this toolkit will assist with:

- Load data from a database and transfer it to a Socrata data portal
- Steps to integrate with an Exchange server to provide e-mail alerts on the outcome of ETL scripts
- Handles deployment issues when using multiple operating systems during development
- Utilities to allow administrators to quickly analyze the log files of ETLs for quick diagnostics

The ETL framework is organized so each function can be modified in one file that is used by all ETLs. This provides for easier maintenance, upgrading, and modification over hundreds of ETLs.

CHAPTER 5

Features

- Open source at the core - this framework can be deployed using Kettle, an open-source ETL software. Pentaho also provides telephone support and training if desired.
- Compatible with multiple data sources - this ETL framework can be used with a variety of data sources, including a range of databases (MySQL, PostgreSQL, Oracle, SQL Server, and variety of NoSQL), APIs, text files, etc.
- Compatible workflow for multiple operating systems - ETLs can be developed and deployed across multiple operating systems. ETLs can be developed on a Windows environment and deployed on Linux
- Helpful utilities - includes several scripts to help users quickly analyze log files

CHAPTER 6

Requirements

The requirements for the recommended configuration require the following pieces of software:

- Kettle (or Pentaho) data integration
- Java 1.6 or higher
- DataSync (for use with Socrata)

CHAPTER 7

Errors / Bugs

Experiencing issues with the included files? Report it on our issue tracker