

Data Load CLI

Schematable loading command-line utility to run in synchronous and asynchronous modes

Goal

The goal of this modification is to make the script generic enough to integrate with third-party tools to schedule loading jobs. The script should work both in synchronous and asynchronous modes.

The script should be able to log status codes (indicating success or failure of the process) so tools using this utility should be able to take appropriate actions based on these status codes.

Synchronous Mode

The control is returned only after complete execution of the load job with the ability to redirect output to a file.

Asynchronous Mode

The control returns immediately after launching the load job with the ability to redirect output to a file.

Usage

```
./load_schema.sh [-h] [-i <incorta-instance-url>] [-n <tenant-name>] [-u <tenant-username>] [-p <tenant-password>] -l <load-method> [-m <input-file-path> / -s <schema-name>] [-t <table-name>] [-f <output-file-path>]
```

Mandatory Arguments

-l <load-method>	One Method from {load_schema, load_schema_incremental, load_schema_from_staging}
-m <input-file-path> / -s <schema-name>	Input file path containing lists of schemas/tables or schema name e.g. Sales

Optional Arguments

-i <incorta-instance-url>	Incorta instance URL e.g. http://127.0.0.1:8080/incorta (If specified it will overwrite the value hardcoded in the script)
-n <tenant-name>	Tenant name e.g. demo (If specified it will overwrite the value hardcoded in the script)
-u <tenant-username>	Tenant username e.g. admin (If specified it will overwrite the value hardcoded in the script)
-p <tenant-password>	Tenant password e.g. admin (If specified it will overwrite the value hardcoded in the script)
-t <table-name>	Table name e.g. Products (Should not be specified while using load_schema_incremental)
-f <output-file-path>	File path for output redirection

Other Options

-h	Display help
----	--------------

Output Redirection

The script should provide the status of the load job either on the screen or should be able to redirect it to a file based on the parameter passed.

In synchronous mode, If the output file path parameter is provided then the script should log output to a file and not return until the load is completed.

In asynchronous mode, If the output file path parameter is provided then the script should log output to a file and return immediately after launching the loading job.

Status Codes

The script should return the following status codes (in the form of integers) at the end of the process, This will enable tools using this utility to take appropriate actions.

Status code is logged at the end of the output as follows:

```
Status=<status-code>
```

Status code is also used as the exit code for the shell script.

Available status codes are:

- 0 - Loading finished successfully
- 1 - Misuse error (before loading is initiated)
- 2 - Login failed (before loading is initiated)
- 3 - Input schema doesn't exist error
- 4 - Input table doesn't exist error
- 5 - Loading finished with errors
- 6 - Loading error
- 7 - Loading interrupted
- 8 - Out of memory error

Examples

Schema Full Load in Synchronous Mode Without Output Redirection

```
./load_schema.sh -l load_schema -s Sales
```

Output (in console)

```
Logging in ...
```

```
Logged in successfully
```

```
Start schema loading ...
```

```
Starting loading job for schema: Sales ...
```

```
Polling loading status ...
```

```
Loading job finished successfully!
```

```
Loading job start timestamp: <formatted-epoch-timestamp>
```

```
Tables
```

```
-----
```

```
Sales.COUNTRIES
```

```
-- State: Loading Finished
```

```
-- Load Type: Full
```

```
-- Extracted Rows: 24
```

```
-- Rejected Rows: 0
```

```
-- Loaded Rows: 24
```

```
-- Extract Duration: 0.0 seconds
```

```
-- Load Duration: 0.0 seconds
```

```
Sales.CUSTOMERS
```

```
-- State: Loading Finished
```

```
-- Load Type: Full
```

```
-- Extracted Rows: 55500
```

```
-- Rejected Rows: 0
```

```
-- Loaded Rows: 55500
-- Extract Duration: 1.0 seconds
-- Load Duration: 0.0 seconds
```

Sales.Months

```
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 12
-- Rejected Rows: 0
-- Loaded Rows: 12
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds
```

Sales.PRODUCTS

```
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 71
-- Rejected Rows: 0
-- Loaded Rows: 71
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds
```

Sales.SALES1

```
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 918843
-- Rejected Rows: 0
-- Loaded Rows: 918843
-- Extract Duration: 8.0 seconds
-- Load Duration: 0.0 seconds
```

Sales.Sales_Date_Dim

```
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 4017
-- Rejected Rows: 0
-- Loaded Rows: 4017
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds
```

```
Loading job finish timestamp: <formatted-epoch-timestamp>
Status=0
```

```
Schema loading finished successfully!
Logged out
```

```
./load_schema.sh -n demo -u admin -p admin -l load_schema -s Sales
```

Output (in console)

```
Logging in ...
```

```
Logged in successfully
```

```
Start schema loading ...
```

```
Starting loading job for schema: Sales ...
```

```
Polling loading status ...
```

```
Loading job finished successfully!
```

```
Loading job start timestamp: <formatted-epoch-timestamp>
```

```
Tables
```

```
-----
```

```
Sales.COUNTRIES
```

```
-- State: Loading Finished
```

```
-- Load Type: Full
```

```
-- Extracted Rows: 24
```

```
-- Rejected Rows: 0
```

```
-- Loaded Rows: 24
```

```
-- Extract Duration: 0.0 seconds
```

```
-- Load Duration: 0.0 seconds
```

```
Sales.CUSTOMERS
```

```
-- State: Loading Finished
```

```
-- Load Type: Full
```

```
-- Extracted Rows: 55500
```

```
-- Rejected Rows: 0
```

```
-- Loaded Rows: 55500
```

```
-- Extract Duration: 1.0 seconds
```

```
-- Load Duration: 0.0 seconds
```

```
Sales.Months
```

```
-- State: Loading Finished
```

```
-- Load Type: Full
```

```
-- Extracted Rows: 12
```

```
-- Rejected Rows: 0
```

```
-- Loaded Rows: 12
```

```
-- Extract Duration: 0.0 seconds
```

```
-- Load Duration: 0.0 seconds
```

```
Sales.PRODUCTS
```

```
-- State: Loading Finished
```

```
-- Load Type: Full
```

```
-- Extracted Rows: 71
```

```
-- Rejected Rows: 0
```

```
-- Loaded Rows: 71
```

```
-- Extract Duration: 0.0 seconds
```

```
-- Load Duration: 0.0 seconds

Sales.SALES1
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 918843
-- Rejected Rows: 0
-- Loaded Rows: 918843
-- Extract Duration: 8.0 seconds
-- Load Duration: 0.0 seconds

Sales.Sales_Date_Dim
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 4017
-- Rejected Rows: 0
-- Loaded Rows: 4017
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds
-----
Loading job finish timestamp: <formatted-epoch-timestamp>
Status=0

Schema loading finished successfully!
Logged out
```

Note: Incorta URL, tenant, username, and password can be hardcoded at the beginning of the shell script part of the tool or it can be provided as command-line arguments as shown in the previous example.

Schema Full Load in Synchronous Mode With Output Redirection

```
./load_schema.sh -l load_schema -s Sales -f load.log
```

Output (in console)

```
Logging in ...
Logged in successfully
Start schema loading ...
Schema loading finished successfully!
Logged out
```

Output (in load.log)

```
Starting loading job for schema: Sales ...
```

```
Polling loading status ...
```

```
Loading job finished successfully!
```

Loading job start timestamp: <formatted-epoch-timestamp>

Tables

Sales.COUNTRIES

-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 24
-- Rejected Rows: 0
-- Loaded Rows: 24
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds

Sales.CUSTOMERS

-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 55500
-- Rejected Rows: 0
-- Loaded Rows: 55500
-- Extract Duration: 1.0 seconds
-- Load Duration: 0.0 seconds

Sales.Months

-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 12
-- Rejected Rows: 0
-- Loaded Rows: 12
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds

Sales.PRODUCTS

-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 71
-- Rejected Rows: 0
-- Loaded Rows: 71
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds

Sales.SALES1

-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 918843
-- Rejected Rows: 0
-- Loaded Rows: 918843
-- Extract Duration: 8.0 seconds
-- Load Duration: 0.0 seconds

Sales.Sales_Date_Dim

-- State: Loading Finished

```
-- Load Type: Full
-- Extracted Rows: 4017
-- Rejected Rows: 0
-- Loaded Rows: 4017
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds
-----
Loading job finish timestamp: <formatted-epoch-timestamp>
Status=0
```

Schema Full Load in Synchronous Mode With Error

```
./load_schema.sh -l load_schema -s Presales

Output (in console)

Logging in ...
Logged in successfully
Start schema loading ...

The following schema does not exist: Presales
Status=3

Schema loading failed!
Logged out
```

Schema Full Load in Asynchronous Mode With Output Redirection

```
./load_schema.sh -l load_schema -s Sales -f load.log &

Output is the same as the second example the only difference is that
the control is returned immediately
```

Table Full Load in Synchronous Mode Without Output Redirection


```
./load_schema.sh -l load_schema -s Sales -t SALES1
```

Output (in console)

```
Logging in ...
```

```
Logged in successfully
```

```
Start table loading ...
```

```
Starting loading job for table: Sales.SALES1 ...
```

```
Polling loading status ...
```

```
Loading job finished successfully!
```

```
Loading job start timestamp: <formatted-epoch-timestamp>
```

```
Tables
```

```
-----
```

```
Sales.SALES1
```

```
-- State: Loading Finished
```

```
-- Load Type: Full
```

```
-- Extracted Rows: 918843
```

```
-- Rejected Rows: 0
```

```
-- Loaded Rows: 918843
```

```
-- Extract Duration: 8.0 seconds
```

```
-- Load Duration: 0.0 seconds
```

```
-----
```

```
Loading job finish timestamp: <formatted-epoch-timestamp>
```

```
Status=0
```

```
Table loading finished successfully!
```

```
Logged out
```

Table Full Load in Synchronous Mode With Output Redirection

```
./load_schema.sh -l load_schema -s Sales -t SALES1 -f load.log
```

Output (in console)

```
Logging in ...
Logged in successfully
Start table loading ...
Table loading finished successfully!
Logged out
```

Output (in load.log)

```
Starting loading job for table: Sales.SALES1 ...
```

```
Polling loading status ...
```

```
Loading job finished successfully!
Loading job start timestamp: <formatted-epoch-timestamp>
```

```
Tables
```

```
-----
```

```
Sales.SALES1
```

```
-- State: Loading Finished
```

```
-- Load Type: Full
```

```
-- Extracted Rows: 918843
```

```
-- Rejected Rows: 0
```

```
-- Loaded Rows: 918843
```

```
-- Extract Duration: 8.0 seconds
```

```
-- Load Duration: 0.0 seconds
```

```
-----
```

```
Loading job finish timestamp: <formatted-epoch-timestamp>
```

```
Status=0
```

Table Full Load in Asynchronous Mode With Output Redirection

```
./load_schema.sh -l load_schema -s Sales -t Product_Sales -f load.log &
```

Output is the same as the previous example the only difference is that the control is returned immediately

Table Full Load in Synchronous Mode With Error

```
./load_schema.sh -l load_schema -s Sales -t Quarters
```

Output (in console)

```
Logging in ...
```

```
Logged in successfully
```

```
Start table loading ...
```

```
The following table does not exist: Sales.Quarters
```

```
Status=4
```

```
Table loading failed!
```

```
Logged out
```

Multiple Schemas/Tables Full Load in Synchronous Mode Without Output Redirection

```
./load_schema.sh -l load_schema -m input
```

Input File (input) Content

```
Sales
```

```
HR.Employees
```

Output (in console)

```
Logging in ...
```

```
Logged in successfully
```

```
Start schemas/tables loading ...
```

```
Starting loading job for schema: Sales ...
```

```
Starting loading job for table: HR.Employees ...
```

```
Polling loading status ...
```

```
Loading job finished successfully!
```

```
Loading job start timestamp: <formatted-epoch-timestamp>
```

```
Tables
```

```
-----
```

```
Sales.COUNTRIES
```

```
-- State: Loading Finished
```

```
-- Load Type: Full
```

```
-- Extracted Rows: 24
```

```
-- Rejected Rows: 0
```

```
-- Loaded Rows: 24
```

```
-- Extract Duration: 0.0 seconds
```

```
-- Load Duration: 0.0 seconds
```

Sales.CUSTOMERS

-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 55500
-- Rejected Rows: 0
-- Loaded Rows: 55500
-- Extract Duration: 1.0 seconds
-- Load Duration: 0.0 seconds

Sales.Months

-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 12
-- Rejected Rows: 0
-- Loaded Rows: 12
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds

Sales.PRODUCTS

-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 71
-- Rejected Rows: 0
-- Loaded Rows: 71
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds

Sales.SALES1

-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 918843
-- Rejected Rows: 0
-- Loaded Rows: 918843
-- Extract Duration: 9.0 seconds
-- Load Duration: 0.0 seconds

Sales.Sales_Date_Dim

-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 4017
-- Rejected Rows: 0
-- Loaded Rows: 4017
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds

Loading job finish timestamp: <formatted-epoch-timestamp>
Status=0

Loading job finished successfully!

Loading job start timestamp: <formatted-epoch-timestamp>

```
Tables
-----
HR.Employees
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 107
-- Rejected Rows: 0
-- Loaded Rows: 107
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds
-----
Loading job finish timestamp: <formatted-epoch-timestamp>
Status=0

Status=0
Schemas/tables loading finished successfully!
Logged out
```

Multiple Schemas/Tables Full Load in Synchronous Mode With Output Redirection

```
./load_schema.sh -l load_schema -m input -f load.log

Input File (input) Content
Sales
HR.Employees

Output (in console)
Logging in ...
Logged in successfully
Start schemas/tables loading ...
Schemas/tables loading finished successfully!
Logged out

Output (in load.log)

Starting loading job for schema: Sales ...

Starting loading job for table: HR.Employees ...

Polling loading status ...

Loading job finished successfully!
Loading job start timestamp: <formatted-epoch-timestamp>
Tables
-----
Sales.COUNTRIES
-- State: Loading Finished
-- Load Type: Full
```

```
-- Extracted Rows: 24
-- Rejected Rows: 0
-- Loaded Rows: 24
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds
```

Sales.CUSTOMERS

```
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 55500
-- Rejected Rows: 0
-- Loaded Rows: 55500
-- Extract Duration: 1.0 seconds
-- Load Duration: 0.0 seconds
```

Sales.Months

```
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 12
-- Rejected Rows: 0
-- Loaded Rows: 12
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds
```

Sales.PRODUCTS

```
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 71
-- Rejected Rows: 0
-- Loaded Rows: 71
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds
```

Sales.SALES1

```
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 918843
-- Rejected Rows: 0
-- Loaded Rows: 918843
-- Extract Duration: 9.0 seconds
-- Load Duration: 0.0 seconds
```

Sales.Sales_Date_Dim

```
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 4017
-- Rejected Rows: 0
-- Loaded Rows: 4017
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds
```

```
-----
Loading job finish timestamp: <formatted-epoch-timestamp>
Status=0

Loading job finished successfully!
Loading job start timestamp: <formatted-epoch-timestamp>
Tables
-----
HR.Employees
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 107
-- Rejected Rows: 0
-- Loaded Rows: 107
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds
-----
Loading job finish timestamp: <formatted-epoch-timestamp>
Status=0

Status=0
```

Multiple Schemas/Tables Full Load in Synchronous Mode With Errors

```
./load_schema.sh -l load_schema -m input

Input File (input) Content
Sales
HR.Employees
Presales

Output (in console)
Logging in ...
Logged in successfully
Start schemas/tables loading ...

Starting loading job for schema: Sales ...

Starting loading job for table: HR.Employees ...

Polling loading status ...

Loading job finished successfully!
Loading job start timestamp: <formatted-epoch-timestamp>
Tables
-----
Sales.COUNTRIES
-- State: Loading Finished
```

```
-- Load Type: Full
-- Extracted Rows: 24
-- Rejected Rows: 0
-- Loaded Rows: 24
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds
```

Sales.CUSTOMERS

```
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 55500
-- Rejected Rows: 0
-- Loaded Rows: 55500
-- Extract Duration: 1.0 seconds
-- Load Duration: 0.0 seconds
```

Sales.Months

```
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 12
-- Rejected Rows: 0
-- Loaded Rows: 12
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds
```

Sales.PRODUCTS

```
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 71
-- Rejected Rows: 0
-- Loaded Rows: 71
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds
```

Sales.SALES1

```
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 918843
-- Rejected Rows: 0
-- Loaded Rows: 918843
-- Extract Duration: 9.0 seconds
-- Load Duration: 0.0 seconds
```

Sales.Sales_Date_Dim

```
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 4017
-- Rejected Rows: 0
-- Loaded Rows: 4017
-- Extract Duration: 0.0 seconds
```



```

-- Load Duration: 0.0 seconds
-----
Loading job finish timestamp: <formatted-epoch-timestamp>
Status=0

Loading job finished successfully!
Loading job start timestamp: <formatted-epoch-timestamp>
Tables
-----
HR.Employees
-- State: Loading Finished
-- Load Type: Full
-- Extracted Rows: 107
-- Rejected Rows: 0
-- Loaded Rows: 107
-- Extract Duration: 0.0 seconds
-- Load Duration: 0.0 seconds
-----
Loading job finish timestamp: <formatted-epoch-timestamp>
Status=0

The following schema does not exist: Presales
Status=3

Status=3 (represents the most common unsuccessful (or 0 in case of no
failures at all) loading status code reported)
Schemas/tables loading finished successfully!
Logged out

```

Note

For asynchronous mode, Please append '&' after the command. This will ensure that the script runs in the background giving control to the user immediately after launching the loading job.

Code

The utility consists of the following scripts:

1. load_schema_CLI_V1.5.sh - Shell script to call Incorta python API
2. incorta_v1.7.py - Modified Incorta python API

The utility does the following steps:

1. Get schema identifier using **schema/getByName** endpoint.
2. Get schema initial status (to check if the schema is currently loading or queued for loading) using **schema/getSchemaStatus** endpoint.
3. If the **schema is neither loading nor queued for loading**, launch a loading job using **schema/loadData** endpoint.
4. If the **schema is loading or is queued for loading, no other job will be launched**.
5. Poll schema status for (successful loading or failure due to errors) using the same endpoint used in step no. 2, **regardless of launching a loading job or not**.
6. Get the details of the latest loading job using **schema/getSchemaJobDetails** endpoint then use these details to show some stats.
7. Terminate with corresponding status code.

The initial version of this utility was developed by Nithish. Changes introduced to the previous version (load_schema_CLI_V1.4.sh & incorta_v1.6.py) are:

1. Fix bug with synchronous mode.
2. Fix bug with python versions compatibility (python 2.7 and 3.7).
3. Change how arguments are passed to the shell script.
4. Add a command-line option to show the utility description, how to use it, and explain each argument that could be passed to it.
5. Use different status codes for robust error handling and meaningful feedback.
6. Modify how loading job stats are reported to provide more info to the user as well as fix bug with reported rows count in case of incremental loading.
7. Support loading multiple schemas/tables using the same command given their names in the input file as follows:

```
SCHEMA_1
SCHEMA_2 . TABLE_1
SCHEMA_3
```

Noting that loading multiple tables from the same schema is not guaranteed to work as intended since the script first checks if the current schema is loading or not if yes it will not wait to issue another loading command instead it will wait to report the loading status and stats.

