
Cloud Elements Formula Migration and Updation Tool

25th September 2019

OVERVIEW

The tool created with Node Js aims at seamless migration or updation of formulas from one cloud elements account to another. With simple commands the formulas are migrated to the corresponding accounts within no time thus helping to the tedious task of doing the same manually.

FUNCTIONALITIES

1. Migration - Migrate a formula that is available in the source account but not created in your destination account.
2. Updation - Update the latest changes made in formulas in the source account (say Development account) to a formulas in the target account (say Destination account).
3. Reversion - Revert any unnecessary changes made in the formula in the target account by selecting the required restore point.

CONFIGURATION

1. This tool works by implementing several http requests. It runs on your local PC and requires the domain name (say **api.cloud-elements.com**), the **user** and **organisation secret** as the basic credentials to access the corresponding formulas.
2. These details are to be provided separately for the source and destination accounts which are distinctly mentioned as **dev** and **prod** keys within the **Input.json** file. Also requirements include the source *Formula ID* for migration (and the destination *Formula ID* in case of updation and reversion). For instance the following example shows the dev key in the Input.json file.

```

"dev": {
  "options": {
    "hostname": "api.cloud-elements.com",
    "headers": {
      "Authorization": "User
/AY/QjevBKxPKAUC17lsNQC0/DLw0d0mdq+yBzilfiU
=, Organization c9584f9cc4a4b4c91568438a9a3041a2",
      "accept": "application/json"
    }
  },
  "Elements": {
    "element_ids": []
  },
  "formulas": {
    "formula_ids": [
      25218
    ]
  }
},

```

3. Also make sure that you edit the **Readinputfile.js** file where it is required to provide the proper directory path as the following example illustrates.

```

const fs = require('fs');
var input;
function Readconfig(fileName){
  return new Promise((resolve,reject) => {
    fs.readFile('/home/prharikamath/sandbox/CloudElementsMigration/CloudElements-Mi
gration/'+fileName,'utf8',(err, contents) => {
      if(err) {
        console.error(err)
        reject(err);
      }
      resolve(JSON.parse(contents));
    });
  })
}
module.exports = Readconfig;

```

INSTALLATION

This tool works only with `node js`. To install node js run the following command in the terminal :

```
$ sudo apt-get update
$ sudo apt-get install nodejs
```

You are expected to have installed `npm`, which is the Node.js package manager. If it is not installed run the following command in the terminal:

```
$ sudo apt-get install npm
```

To install the tool run the following command in the terminal:

```
$ npm install
```

❖ **NOTE** : npm version 6 or above preferred

COMMANDS

```
$ npm run migrate
$ npm run update
$ npm run revert
```

NOTE

The user has to ensure that he/she types in the required Formula IDs everytime he/she runs the tool for updation (works with multiple Formula IDs simultaneously. Just type them down as the array elements in the **formula_ids** object key inside **Input.json** file. The **mapper** object in Input.json file updates itself after every new **migration** step where it stores the source Formula ID as the key and the destination Formula ID as the value. Mapper can thus be used for later **updation** operations (not considered for new updation requests).