# Tricentis – Configurations Migration Utility Scripts

Below are the full instructions on how to leverage the export/import scripts.

Preparation:

All scripts are located here: <https://github.com/qaschris/legacy-configuration-migration/tree/main>  
The latest version of Node.js must be installed on the hosts where the scripts will be executed.  The export scrip may be run on any workstation with direct qTest database access. The import scripts may be run on any workstation with access to the qTest Manager API via HTTPS.  The scripts to be executed are all js files, but I have included the raw SQL file in case the SQL queries need to be validated on Dell’s system prior to script execution.

Step 1 - Configurations Export:

Usage: node export\_legacy\_configurations.js

Configuration:  Update the below values in the script with those used by the qTest user or a postgres db user with rights to read the qTest schema.  Also, add the qTest project IDs delimited with commas and spaces to the ‘projectIds’ constant.

Prerequisites: You may need to run ‘npm install pg’ at the command line prior to script execution.

const pool = new Pool({

  user: 'qtestadmin',

  host: 'localhost',

  database: 'qTest',

  password: 'qtestadmin',

  port: 5432, // Default PostgreSQL port

});

const projectIds = '49, 54, 61' // Replace these with comma delimited qTest project IDs to filter based on project, ex. '49, 54, 61'

The resulting execution will export three JSON files containing variable, configuration and test run data.  Open these in any text editor and manually verify the data exists and is accurate to the best of your ability.

Step 2 - Variables Import:

Usage: node import\_variables.js

Configuration: Update the below values in the script with the Bearer token and qTest domain name for the instance to which you’ll be importing the data, as well as the qTest Project IDs to which you wish to give access to these variables.

Prerequisites: You may need to run ‘npm install axios’ at the command line prior to script execution.

const apiEndpoint = 'https://qteststaging2.staging.qtestnet.com/api/v3/variables';

const bearerToken = 'aa718c08-e087-4ad3-ad0c-831fc106677e';  
const projectIds = [96134];  // This is an array of project IDs for which you want the configurations to appear, ex. [900, 901, 910]

The resulting execution will import the variables and create a JSON file named variable\_id\_lookup.json which will contain a lookup table of legacy versus new variable ids that will be used in the next step.  Please log into qTest and verify in the new configurations feature that the variables were successfully imported.

Step 3 - Configurations Import:

Usage: node import\_configurations.js

Configuration: Update the below values in the script with the Bearer token and qTest domain name for the instance to which you’ll be importing the data, as well as the qTest Project IDs to which you wish to give access to these variables.

Prerequisites: You may need to run ‘npm install axios’ at the command line prior to script execution.

const apiEndpoint = 'https://qteststaging2.staging.qtestnet.com/api/v3/configurations';

const bearerToken = 'aa718c08-e087-4ad3-ad0c-831fc106677e';  
const projectIds = [96134];  // This is an array of project IDs for which you want the configurations to appear, ex. [900, 901, 910]

The resulting execution will import the configurations using the variable\_id\_lookup.json and create a JSON file named configuration\_id\_lookup.json.  The file will contain a lookup table of legacy versus new configuration ids that will be used in the next step.  Please log into qTest and verify in the new configurations feature that the configurations were successfully imported with the correct variable information.

Step 4 - Test Runs Update:

Usage: node update\_testruns.js

Configuration: Update the below values in the script with the Bearer token and qTest domain name for the instance for which you’ll be updating the data.  You will not need to configure project IDs as these are brought in from the export process already.

Prerequisites: You may need to run ‘npm install axios’ at the command line prior to script execution.

const apiEndpoint = 'https://qteststaging2.staging.qtestnet.com/api/v3/projects/';

const bearerToken = 'aa718c08-e087-4ad3-ad0c-831fc106677e';

This is the final script and should only be run AFTER all other scripts have been successfully executed and the data 100% validated in the UI.  The resulting execution will update the test runs with the associated configurations by using the configuration\_id\_lookup.json file created in the last step, and the testruns.json from the first step.  Data may be validated in qTest afterwards. Note: Execution History (aka Test Logs) will not be updated as this is an audit history of prior executions.