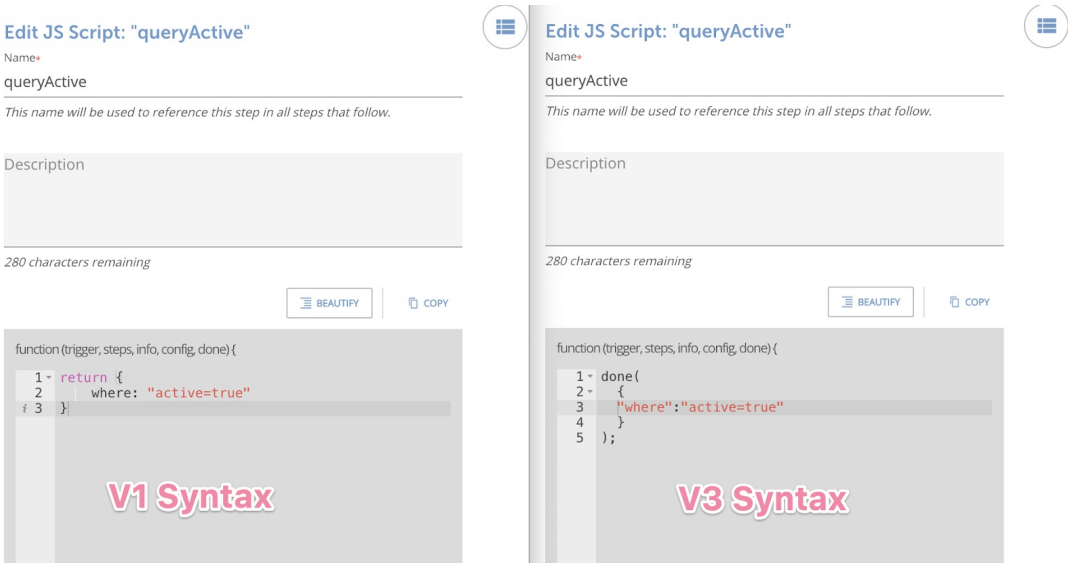
Upgrading a Formula to the V3 Engine

Last Modified on 07/13/2021 10:46 am EDT

All existing customers with V1 formula engine usage have been reached out to individually regarding this information. We are nearing the end of life for the V1 formula engine and will remove support for it at the end of Q2, 2021. Although SAP Open Connectors released the V3 formula engine in 2017, we continued supporting v1 formulas for the last few years, but are now officially sunsetting the v1 engine as of June 30, 2021. Note that all formula-related documentation in the knowledgebase is based on the v3 formula engine.

Steps to follow:

1. Export existing V1 formulas from your production account. For more information, refer to Import and Export Formula Templates.
2. Import the V1 formulas into your staging account. For more information, refer to Import and Export Formula Templates.
3. Review the design of your V1 formulas to determine if any of the following conditions apply:
    - Does your formula include a Notification Step?
        - **Notification Step** is no longer supported; if you do have a notification step, you must remove the step and replace it with a **JS Script** type step and use `notify.email('email address','subject','body')` to send a notification email.
    - Are there any JS Script steps that use `return` rather than `done()`?
        - The V3 formula and scripting engine requires every JS Script step to have a `done()`; otherwise, the formula will fail to run. All `return` statements must be replaced with `done()`. For example:



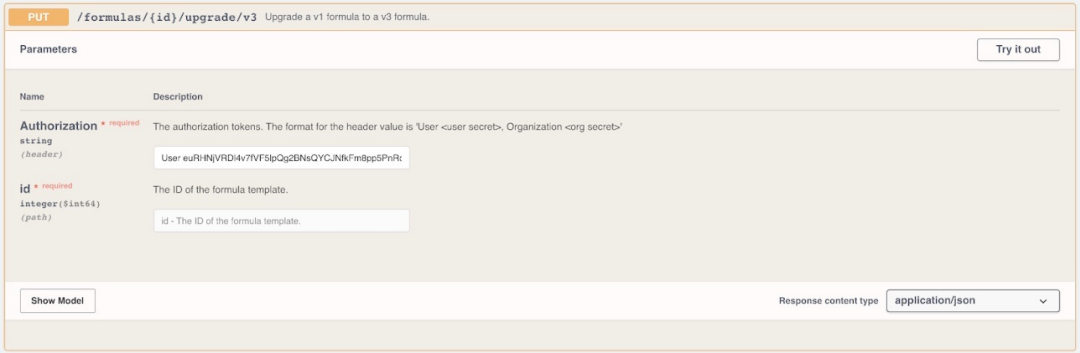
    - Is your formula **Single-Threaded**?
        - The V3 formula engine does not support single-threaded formula executions. If you require single-threaded formula executions, please reach out to our Technical Support team to discuss possible solutions.
    - Anywhere in the formula, is there a reference to any formula context without specifying the object? In other words: do you use formula **context shorthand**? For example:
        - To refer a configuration variable (sourceinstance) are you using `${config.sourceinstance}` or `${sourceinstance}`? `${sourceinstance}` is invalid as it isn't specifying `config`. The same

applies to `trigger`, `steps`, and `info`. Hence, to refer to `trigger`, use `${trigger.}`; when referring to a step, use `${steps.stepname}`, etc.

- Do any Connector and Platform API Request Steps in your formula contain a `Path`?
  - The V3 formula engine upgrade will automatically resolve any existing usage of `Path` and replace it with proper syntax in the `Element Resource URL` for V3 engine processing. For future formula development, ensure that `Path` is not used in any new V3 formula design.
- Do any formula steps reference the Header of a prior Platform or Element Request Step?
  - Check all JS Script steps to ensure that the proper capitalization is referenced. Note that API Request headers in V1 are capitalized, while headers in V3 are not capitalized, eg:
    - V1 headers use initial caps: `Elements-Next-Page-Token`
    - V3 headers are in lowercase: `elements-next-page-token`

4. In Staging, modify your formula steps by following the guidelines from Step 3 if necessary.
5. Upgrade your formulas to the V3 Engine
   - Do NOT simply change your formula engine from V1 to V3 in the UI. Changing the engine is not a comprehensive way to upgrade your formula and could result in issues.
     - Execute the following API (Platform API Documentation Link):

```
 curl --location --request PUT 'https://staging.cloud-elements.com/elements/api-v2
/formulas/{id}/upgrade/v3' \
--header 'accept: application/json' \
--header 'Authorization: User XXXXX, Organization XXXXX' \
--header 'Content-Type: application/json' \
```



6. In staging, test your formulas to ensure that your use case is successful while running on the V3 engine.
7. Upgrade your production formulas to the V3 engine
   - First, locate the staging formula IDs for the formulas that you just validated, and locate the equivalent production formula IDs for integration.
   - Next, `GET /formulas/{id}` in staging to obtain each complete formula template.
   - Take each formula template json and perform a `PUT` to replace the template of the appropriate production formula by ID:

```
curl --location --request PUT 'https://api.cloud-elements.com/elements/api-v2/formulas/
{id}' \
--header 'accept: application/json' \
--header 'Authorization: User XXXXX, Organization XXXXX' \
--header 'Content-Type: application/json' \
```

   - If your formula is in the EU-Production environment, then use the EU-Production endpoint: https://api.cloud-elements.co.uk/elements/api-v2/formulas/{id}