Service Configuration Utility

See @ Tyler.Watson (Unlicensed) for questions relating to this page.

> 📄 **Note**: This page is current with **2022.09**. For information on Service Configuration Utility content and procedures from prior versions, see Archived Service Configuration Utility Procedures and Content.

Overview

The Service Configuration Utility is used to add and update service configurations for each release. Because variables and configurations change with each release, it's vital to have a system that consistently updates these variables and configurations. Users will need to deploy the Utility from Octopus and make changes to the configurations and variable settings that will then allow the Utility to function and input the correct configurations.

> 📄 **Note**: There is no configuration for this utility.

Audience

This document is intended for Release Management and CallMiner developers who use the Service Configuration Utility to add and update new configurations for each release.

The Service Configuration Utility and Related Content

The Update Utility

An automated system used in a repeatable and deterministic way to perform Service Configuration system updates from one version to the next. The Utility automatically applies new settings and variables, ensuring that all Service Configuration settings are current with each release.

███████████ .json

The ███████████ .json file is a JSON configuration file used by the Service Configuration Update Utility during Eureka platform installations and upgrades to populate the ███████████ collection.

For more information, see ███████████ .json.

Service Configuration Application

The Service Configuration application allows you to add and update service configurations via the Service Configuration API, which then stores the configurations in the ███████████ database. The goal of this application is to make it easier for administrators to add, update, and delete configurations stored in the ███████████ database.

For more information, see the Service Configuration User Guide.

How to Run the Service Configuration Utility

> **Note**: Before you run the Service Configuration Utility, make sure that you've deployed CallMiner's Identity Provider and the Service Configuration API.
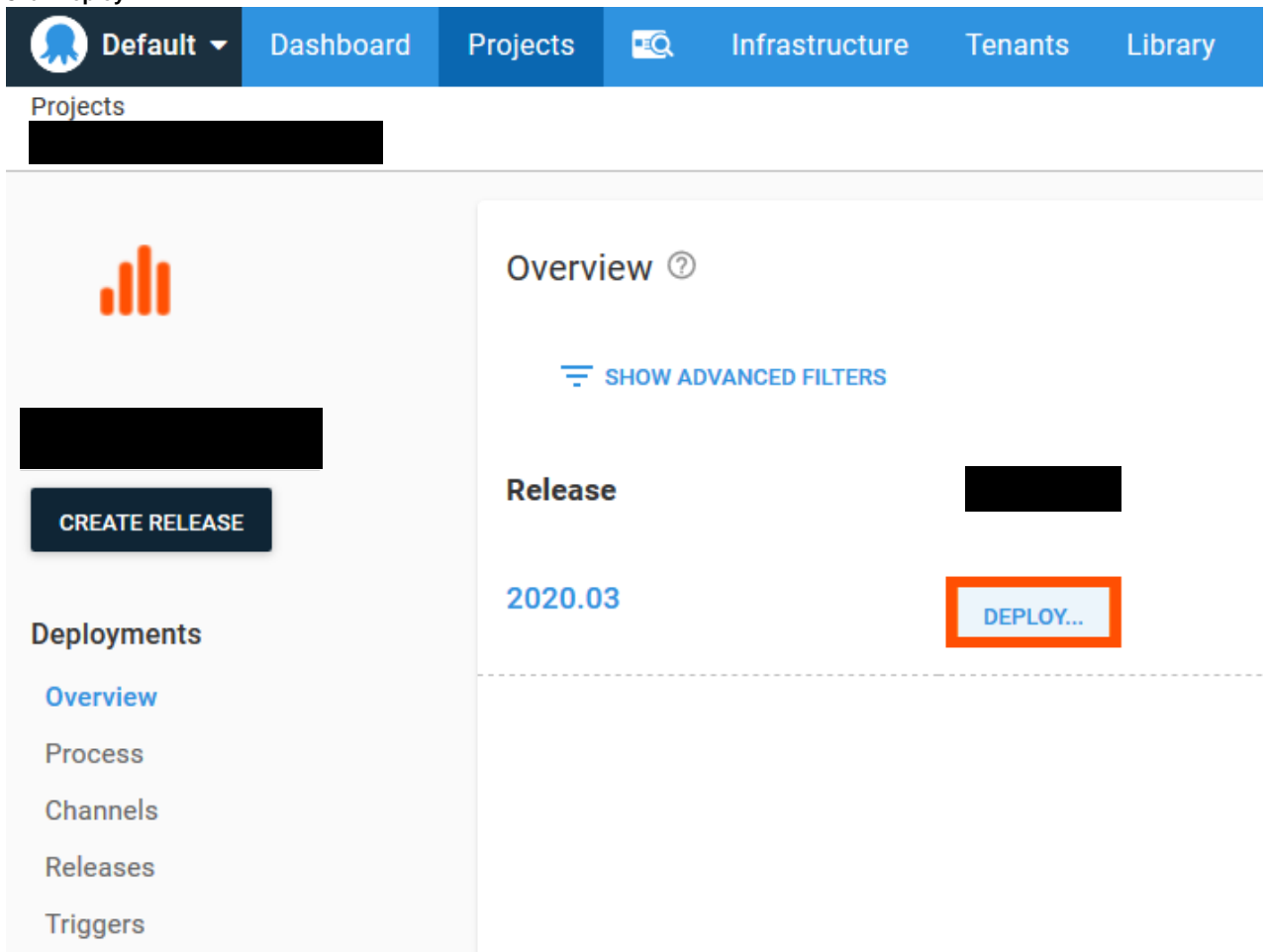
∨ [RM ONLY - Skip this step if you're dev] 1. Deploy the ⬛⬛⬛⬛ Project in Octopus

> ⚠ Skip this step if you are a dev - this is only for RM to do.

1. In the Octopus Deploy Web Portal, search for ⬛⬛⬛⬛ and click the ⬛⬛⬛⬛ project.
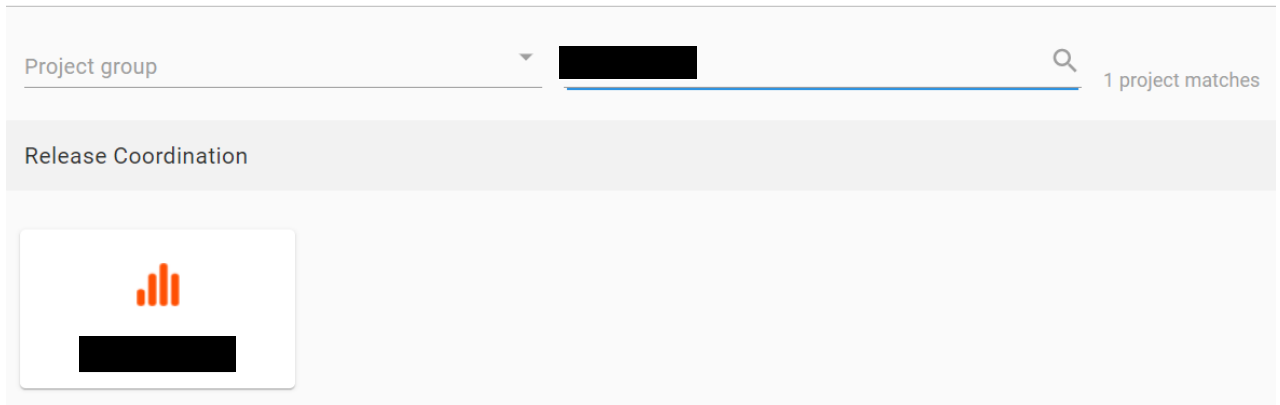
2. Click **Deploy**.

3. Click the green **Deploy** button.

4. As a result, the deployment of the project begins. Monitor the progress of the deployment in **Task Progress**. If the deployment fails, check the **Task Log** to troubleshoot errors and redeploy the project.

5. Verify the success of the deployment. If the deployment fails, check the **Task Log** to troubleshoot errors and redeploy the project.

Release 2021.03

✅ Deploy ███████████ release 2021.03 to ████ ⓘ

TASK SUMMARY   **TASK LOG**

This task started 4 days ago and ran for 7 minutes

⊟ **Deploy** ████████ **release 2021.03 to** ████

The deployment completed successfully.

⊞ **Acquire packages**

⊞ **Step 1: Deploy** ██████████████████

⊞ **Step 2: Deploy** ████████████

⊞ **Step 3: Deploy Service Configuration Utility**

⊞ **Step 4: Deploy** ███████████

⊞ **Apply retention policy on Tentacles**

The ███████████ project is now deployed.
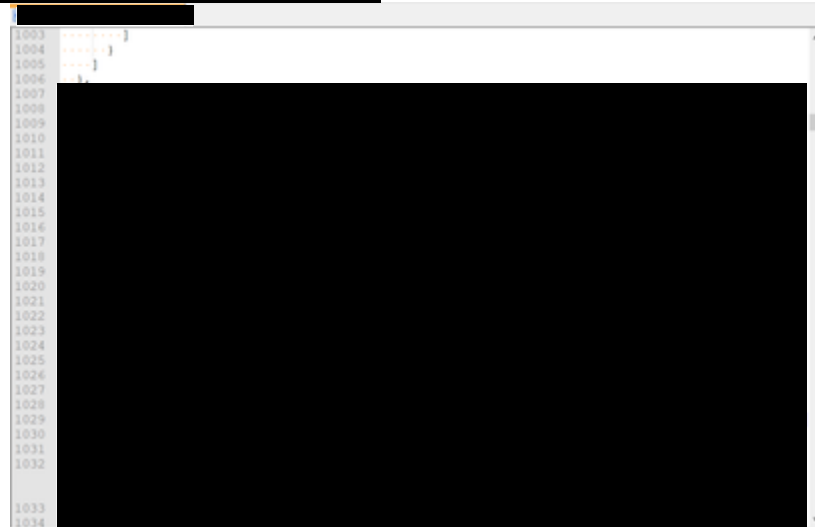
⌄ 2. Modify the ████████.json Template File

The ████████ API fails to deploy if you do not modify the template file. Follow the steps in this section to modify the template file so that deployment of the ████████ API succeeds.

1. Open the deployment location of the Service Configuration Utility directory. By default, this will be the \\███████████████████████ ███ directory on your ████ server.
2. In the ███████████████ folder, edit the ████████████████.**202x** file for this release. The file's **Type** should match the release version of your upgrade. For example, the **Type** for release 2021.03 would be 03 File.

3. Delete the ████████████████████████████████ section.

4. Save the file.

∨ 3. Run the Service Configuration Utility in Preview Mode

> **Note**: The Client ID and Client Secret should already be filled, but if not, follow steps 1-6 to supply the Client ID and Client Secret for the **ClientIdReadWrite** and **ClientSecretReadWrite** values. If they have already been filled, skip to step 7.

1. Return to the Octopus Deploy Web Portal. Search for and select the ███████████████ project.
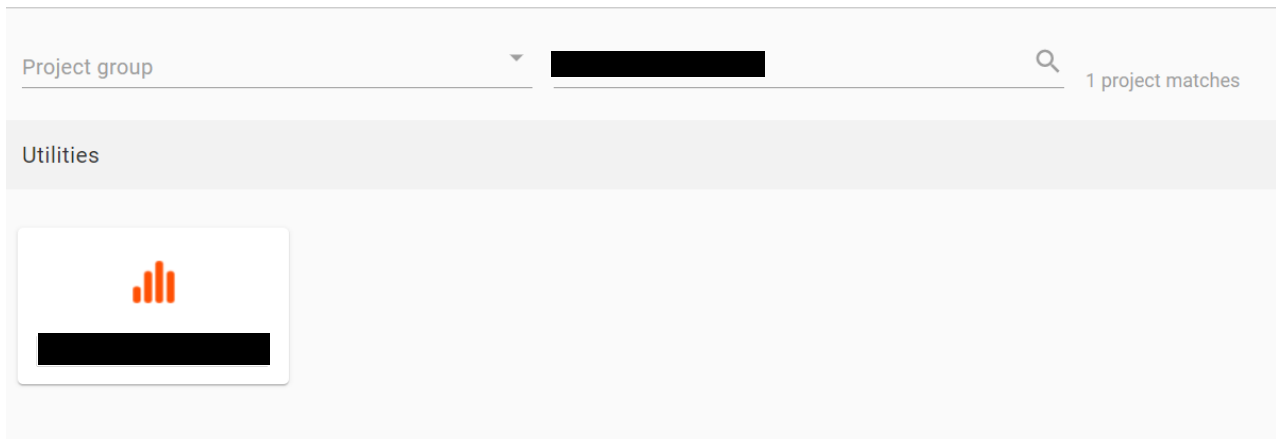
2. Click the latest release deployment.

Overview

SHOW ADVANCED FILTERS

Release

2021.04

2021.04
Apr 28, 2021 11:27 AM

DEPLOY...

3. Click **Task Log**.

Release 2021.04

Deploy ██████████████████████ release 2021.04

TASK SUMMARY   **TASK LOG**

Task Progress

This task started 13 days ago and ran for 3 minutes

✔ Acquire packages
✔ Step 1: Run ████████████████ - Admin
  ✔ ████████
✔ Step 2: Run ████████████████ - Eureka
  ✔ ████████
⚠ Step 3: Run ████████████████ - NGM
  ⚠ ████████
✔ Apply retention policy on Tentacles
  ✔ ████████

4. Copy the credentials.

Release 2021.04

Deploy ██████████████ release 2021.04
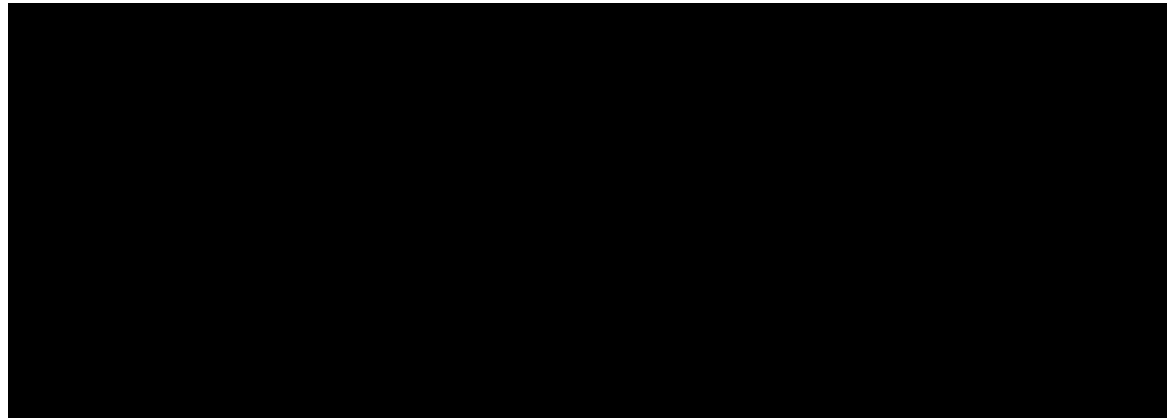
```
Created Client
{
        ClientName: ServiceConfigurationApi.ReadWrite
        Id: 2566
        ClientId:  ████████████████████
        ClientSecret: ███████████████████
}
2021-04-28 11:28:14.736 -04:00 [Information] ████████████████ ran Successfully
2021-04-28 11:28:14.736 -04:00 [Information] Exiting Application
```

5. Click **Library**>**Variable Sets**, then click the **ServiceConfigurationTool** variable set.

Variable Sets
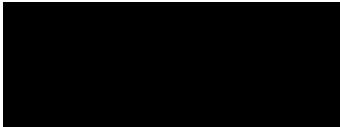
Release Coordination

ServiceConfigurationTool
These variables are for the ServiceConfigurationTool Runbook.

Web Components

6. In the **ClientIdReadWrite** value, enter the Client ID you got from the ███████████████. In the **ClientSecretReadWrite** value, enter the Client Secret you got from the ███████████████, then click **Save**.

Variable Sets

███████████████ ⊙

VARIABLES   VARIABLE TEMPLATES   USAGE   SETTINGS

By variable name   🔍   ⇶ SHOW ADVANCED FILTERS

Name	Value	Scope
Enter new variable	Enter value	Define scope
		ADD ANOTHER V
████████████████████████████		
████████████████	Enter value	Define scope
ClientId	Multiple values	Multiple scopes
	Enter value	Define scope
	Enter value	████████
ClientSecret	Multiple values	Multiple scopes
	Enter value	Define scope
	Enter value	████████

7. Click the **Projects** tab in the Octopus Web Portal, then search for and select **Service Configuration Utility**.

Project group ▾	ServiceConfigurationUtility 🔍	1 project matches

Utilities

ılıı

ServiceConfiguration
Utility

This script runs the
SC_Commands in preview
and update mode. It will be
up to the user to selected

8. Click **Deploy**.

Service Configuration Utility

CREATE RELEASE

Deployments ▾   Operations ▾   Variables ▾   Tasks   Settings

Overview

≡ SHOW ADVANCED FILTERS

Release

2021.05   DEPLOY...

---

9. Choose your environment from the **Select environments** drop-down list.

Service Configuration Utility
Run Snapshot 7WSDJ4X   RUN

Environments   Select one or many environments   ^

                Select environments

                Select environments   ▾

                SHOW ADVANCED

10. Choose **Preview Mode** from the **Mode** drop-down list.

11. When installing Eureka, under **PreviousReleaseVersion** enter *empty*, and under **ReleaseVersion** enter the current version of Eureka that you are installing.

   a. When updating Eureka, enter the previous release version number in the **PreviousReleaseVersion** prompt and then the current release version number in the **ReleaseVersion** prompt.

12. Click **Run** to run the tool.
13. Once the tool has completed its run in preview mode, it prompts you to define the values in the **LIST OF UNRESOLVED TOKENS**. Unresolved tokens are values in the source master.json file that are not configured. These values are indicated this way: **#{TokenName}**.

Service Configuration Utility Runs

✔ Run Service Configuration Utility on ▮▮▮▮

```
⊟ Step 1: Run SC_Command

  ⊟ CMATLADMIN01

    Running the 2021.03 SC_Commands in Preview Mode on ▮▮▮▮

    ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮

    ClientID = ********
    ClientSecret = ********
    Source = ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
    Target = ▮▮▮▮
    2021-02-25 10:42:05.032 -05:00 [Information] Processing please wait...
    2021-02-25 10:42:09.798 -05:00 [Information] FOLLOWING VARIABLES HAVE BEEN ADDED TO ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮ REPOSITORY:
    ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
    PLEASE POPULATE THE VARIABLE ABOVE USING SERVICE CONFIGURATION UI (DATA SEEDING TAB) BEFORE RUNNING THE TOOL IN UPDATE MODE
    ----------------------------------------------------------------------------
    (4) SOURCE SECTIONS CONTAIN ONE OR MORE UNRESOLVED CONFIGURATION TOKENS
    UNRESOLVED TOKENS CAN BE SATISFIED USING ONE OR MANY OF THE FOLLOWING METHODS
    1 - ADDING KEY VALUE ENTRIES INTO THE GLOBAL KEY-VALUE REPLACEMENTS LIST
    2 - ADDING SECTIONS WITH ASSOCIATED VALUES INTO THE SECTION BASED JSON PATH REPLACEMENT LIST
    3 - REMOVING CONFIGURATION TOKENS FROM THE AFFECTED SOURCE SECTION(S) AND REPLACING THEM WITH EXPLICIT VALUES
    LIST OF UNRESOLVED TOKENS
    #▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
    #▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
    #▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
    #▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
    #▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
```

As a result of running the tool, the tool populates the ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮ collection with the variables and default values present in the ▮▮▮▮▮▮▮▮▮▮ .json file. The ▮▮▮▮▮▮▮▮▮▮ .json file is a configuration file used by the Service Configuration Update Utility during Eureka platform installations and upgrades to populate the ▮▮▮▮▮▮▮▮▮▮▮▮ collection. You can now continue to the next section where you'll define these values in the Service Configuration browser application.

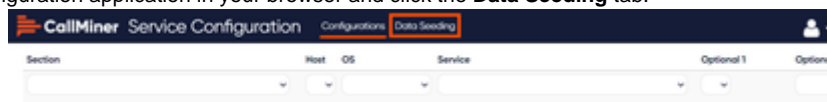⌄ 4. Use the Service Configuration Application to Enter Values for Variables

You can use the Service Configuration Update Utility and Service Configuration browser application to enter the variable values in the Service Configuration database. You can access the Data Seeding tab from the application to modify the updated values for the release. For more information on the Service Configuration application and for a complete list of Service Configuration sections, see the Service Configuration User Guide.

> 🗒 **Note**: For new installations, to determine which values are required, see the **Description** column in the **Data Seeding** tab of the Service Configuration application.
>
> To determine which values need to be entered for upgrades, you can check the ▮▮▮▮▮ .json template in the Utilities folder of your upgrade package to find which sections a variable is used in. Then, click the **Configurations** tab in the Service Configuration browser application and navigate to the sections that have that variable. If a variable already has a value, it does not need to be updated in the **Data Seeding tab**.

Follow these steps to update the Service Configuration database:

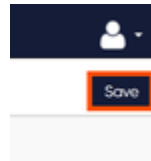1. Open the Service Configuration application in your browser and click the **Data Seeding** tab.

2. If this is your first time using the **Data Seeding** tab, you need to enter the values for any variables with required values and any variables that have not already been configured. If you have previously used the **Data Seeding** tab, you only need to enter the required values for new or updated variables.
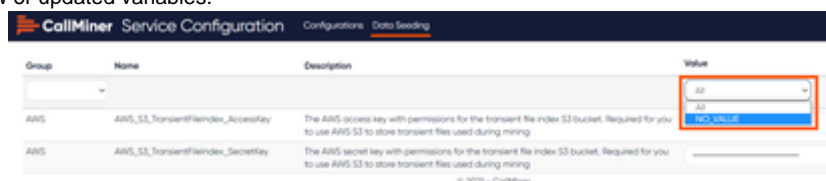
3. Press **Save** to save the changes in the Service Configuration application. The application updates the ▮▮▮▮▮▮▮▮▮▮▮ collection with the new values.

You have now updated the variables with their values for this release.

5. Run the Service Configuration Utility in Update Mode

To finalize updating the variables for the Service Configuration database, you need to run the Service Configuration Utility in Update mode. Follow the steps in this section to run the Service Configuration Utility and update configurations in the Service Configuration database.

1. Return to the Service Configuration Utility project where you ran the Service Configuration Utility in preview mode and click the deployment you ran earlier.
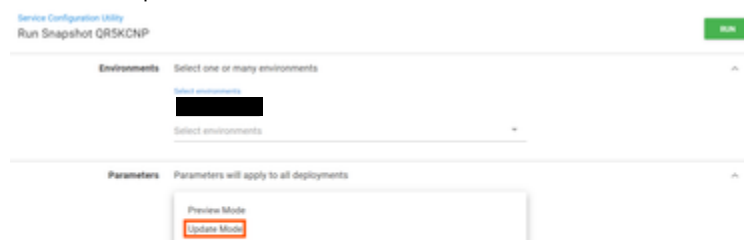

2. Click the three vertical dots next to the **Deploy to...** button and click **Re-deploy....**

3. Choose **Update Mode** from the **Mode** drop-down list.

4. When installing Eureka, under **PreviousReleaseVersion** enter empty, and under **ReleaseVersion** enter the current version of Eureka that you are installing.

Service Configuration Utility

Run Snapshot VXVPR6N

Environments	Select one or many environments
	Select environments
	Select environments ▼

Parameters	Parameters will apply to all deployments
	Mode
	Update Mode ▼
	PreviousReleaseVersion
	empty
	ReleaseVersion
	2021.04

SHOW ADVANCED

a. When upgrading Eureka, enter the previous release version number in the **PreviousReleaseVersion** prompt and then the current release version number in the **ReleaseVersion** prompt.

Service Configuration Utility

Run Snapshot WWV3FY4

RUN

Environments	Select one or many environments	^
	Select environments	
	Select environments ▼	

| Parameters	Parameters will apply to all deployments	^
	Mode	
	Update Mode ▼	
	PreviousReleaseVersion	
	2021.02	
	ReleaseVersion	
	2021.03	

SHOW ADVANCED

5. Click **Run** to run the tool. If the update succeeds, you should see "Changes successfully applied to target system" in the **Task Log**. If you receive an error, verify that you entered the correct values in the Service Configuration application and try again.
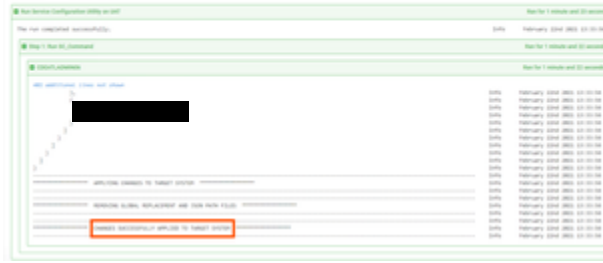
Shared configuration setup is complete.

Miscellaneous

For devs who need to update the configurations for a release, you may need to perform actions in addition to running the Service Configuration Utility. Sometimes, this may include creating new configuration sections or setting a configuration to be an empty string. This section addresses these two particular instances.

Defining and Setting Values for the Variable Replacements

In the event you create a new configuration section and its configuration value has something like a database connection string, you may need to add or modify settings in the ▮▮▮▮▮.json.template file in the Service Configuration Utility. To do so, you'll need to edit the #{VariableName} token within the ▮▮▮▮▮.json.template file. The update utility will not be able to update the settings if there are variables that are not defined or that do not have values. For more information, see Process for Updating Environment Configurations.

Setting an Empty Variable String

At times, you may have unused variable replacements included in the Service Configuration UI. In this instance, you would define the configuration to be an empty string. Follow these steps to set a variable to empty:

1. In the Service Configuration Utility tools folder, open the ▮▮▮▮▮▮.json file in a text editor.
2. Search for the variable value that allows an empty string that you want to modify.
3. Set the value to *true*.
4. In the Service Configuration UI, locate the replacement variable you modified and select its associated value textbox. This textbox is outlined in red.
5. Select a different red-outlined textbox. You'll receive a pop-up confirming whether you want to clear the field of the first replacement variable you clicked.
6. Select **OK**.

The variable value is now empty.

Finding and Updating the ▮▮▮▮▮▮▮▮▮▮.JSON File in Source Control

Go to ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮ in DevOps and open the ▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮ file. You can then edit both the ▮▮▮▮▮▮▮▮▮.json file and the ▮▮▮▮▮▮▮▮▮.json file from the Solution Explorer window.

Troubleshooting

Below are some errors you may encounter when using the Service Configuration Utility. Many errors have to do with the formatting of the JSON, such as typos that would make it impossible to find the ▮▮▮▮▮▮▮▮▮ or ▮▮▮▮▮▮▮▮▮▮ files. "Domain" value typos will also throw errors. However, some errors may occur unrelated to typing errors. These errors have been noted below.

- **"Command line argument, Mode can not be null or empty."**: Displays if the mode in the command is empty. You'll need to provide the missing information.
- **"Multiple sections found in target API. Duplicate section are not permitted."**: Displays if you entered the same section more than once. Remove the duplicate section.
- **"Invalid OS value found."** Displays if the OS property in the json-path-replacement.json array is set to something other than the pre-defined values (Any, Linux, WIndows).
- **"Input file is missing 'sections' section."**: Displays if the json-path-replacement.json file does not contain an array of at least one section object.
- **"Duplicate config section found. Section {sectionName} and Domain {domainName}"**: Displays if you have entered the same section more than once. Remove the duplicate section.
- **"Backup file \"{options.BackupFile}\" not found.")**: Check the spelling of your entry and try again.

- **"Command line argument, Service Configuration MongoDB connection string is invalid."**: Check the spelling of your entry and try again.
- **"Command line argument, SourceJsonFullPath path \"{options.SourceJsonFullPath}\" does not exist."**: Check the spelling of your entry and try again.
- **"Command line argument, TargetJsonFullPath path \"{options.TargetJsonFullPath}\" does not exist."**: Check the spelling of your entry and try again.
- **"Command line argument, TargetApiAuthorityUri = \"{options.TargetApiAuthorityUri}\" is invalid."**: Check the spelling of your entry and try again.
- **"Command line argument, TargetApiBaseUri = \"{options.TargetApiBaseUri}\" is invalid."**: Check the spelling of your entry and try again.
- **"Required parameter \"{((NamedError)error).NameInfo.LongName}\" is missing."**: Check the spelling of your entry and try again.

Related Links

- ███████████████████ Update Utility: Contains rules dedicated to operational scenarios of the Service Configuration Utility.
- Archived Service Configuration Utility Procedures and Content: Contains information related to older versions of the Service Configuration Utility.