SALESFORCE CONFIGURATION VS CUSTOMIZATION ESTIMATOR TOOL

ABSTRACT
Salesforce Configuration vs Customization tool document. Having details about tool, how it works, capabilities and details about manual activities need to do after installing tool in salesforce org.

Gupta, Varun R.
Salesforce Architect

Contents

Introduction

Now a days clients prefer OOB configuration over any custom development in salesforce unless no other option left. Hence, to meet client expectations development team use to do development with OOB configuration. However, some of the client wants to know the amount of configuration and customization happened in per release or at org level.

As there is no OOB tool by which development team nor client can get the stats to calculate amount of configuration and customization. Due to which development team has to calculate manually by running Tooling API on each and every salesforce metadata object | store every tooling API result in excel file | differentiate between configured metadata (ex: Lightning flow, fields, process builder, etc) and customized metadata (ex: LWC, Apex class, etc) | get total and difference | get percentage.

Generally, manual processing of tooling API is very time consuming because of:

- Creation of tooling API query and get result in excel file.
- Segregate Managed and unmanaged records.
- From unmanaged records further segregate custom and config records.
- Do same for **40 - 45** tooling metadata objects.
- Place the final excel and generate a pivot table.

About the tool

This is a salesforce tool basically calculate the percentage of CONFIGURATION & CUSTOMIZATION done in salesforce org and gives a visual and instant result on screen. Tool has flexibility to handle almost all salesforce metadata's.

How tool works

- On trigger of tool, first tool fetches all the configured Tooling sObjects. Ex: Apex class, Trigger, etc...
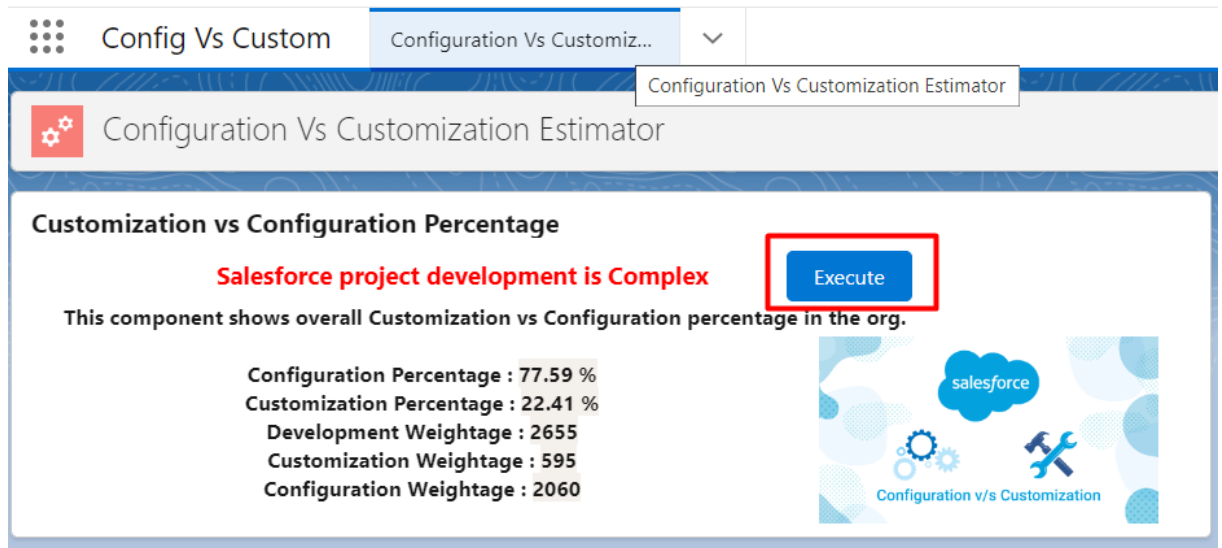
(To navigate on above screen open "Config Vs Custom" console | click on "Configuration Vs Customization Estimator" tab)

- Prepare Tooling queries dynamically and then execute them. Reference link

- Then segregate Managed and Unmanaged metadata from tooling query result.

  - If metadata is Managed (ex: metadata created by Package) then tool will simply count the number of metadata's.

  - If metadata is Unmanaged, then tool will segregate customized & configured unmanaged metadata records.

- For configured metadata: tool will again simply count the number of metadata's.

- For customized metadata: tool will check if metadata is related to APEX CLASS, APEX TRIGGER etc.

  - If yes, then dynamically assign customization weightage (already set by admins in config table) to each record based on code length.

  - If no, then count the number of metadata's.

- Calculate the count and name of all metadata created in last x days. Ex: 7 days

- Store all above stats in custom object | Calculate percentage of CUSTOMIZATION & CONFIGURATION | Show final stats on screen.
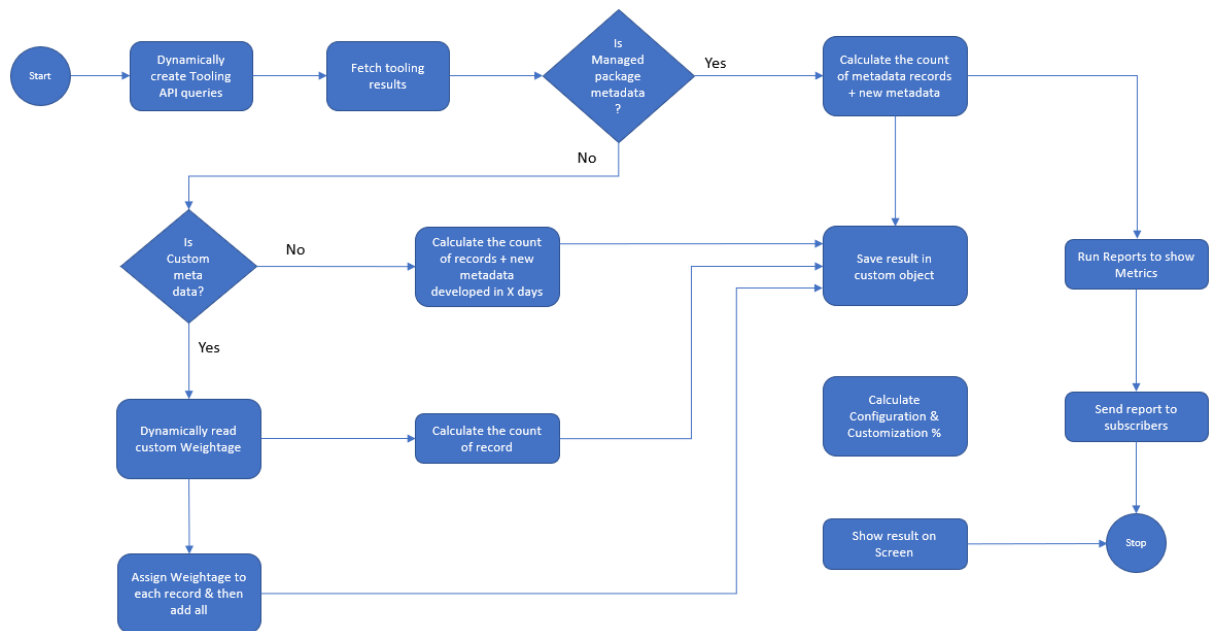
Capabilities of Tool

- Calculate percentage of customization & configuration happened in salesforce org.

- Calculate Customization & Configuration percentage difference between current and last results.

- Show metrics based on release. (i.e show metrics based on date)

- Can show amount of configuration or customization done on which specific object. Ex:

    - 40 custom metadata records (ex: fields, auto response rule, validation rule) on Case object.

    - 1 custom metadata (ex: Trigger) on Account object.

- Facilitate admins who wants to give dynamic customization weightage-based complexity of customization. Ex: If Apex code is of more than 1000 characters then increase the dynamically customization weightage.

- Dynamic Alerts to admins if configuration or customization percentage exceed to admin set threshold limit.

- Reporting on above.

- Subscription to Dashboard.

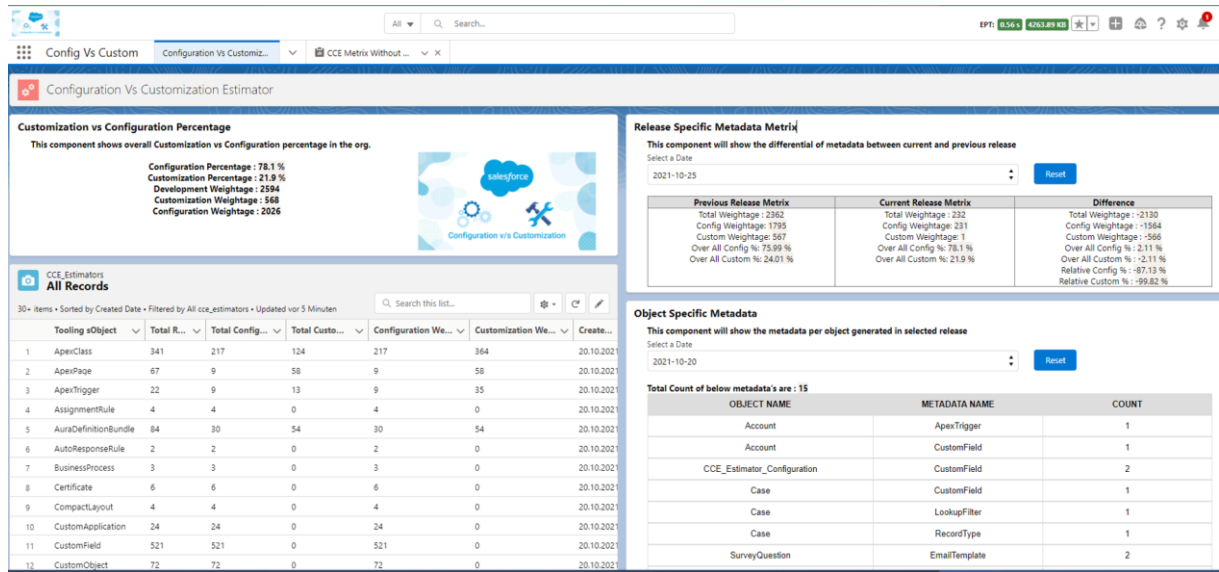- Ability to send manual notifications to target audience.

Target Audience

- Small & large scale client.

- Clients focused on OOB deployment. Like "Sky Deutshland, Munich"

- Best suitable for client those don't have dedicated Metadata migration tools like Copado, Autorabitt etc.

- Clients those who wants to give dynamic customization weightage-based complexity of customization. Ex: if Apex code is of more than 1000 characters then increase the dynamically customization weightage.

- Clients want such results instantly.

- Clients looking for automation to get report above same after every release.

Flow Diagram

```
Start → Dynamically create Tooling API queries → Fetch tooling results → Is Managed package metadata ?
```

Is Managed package metadata ? --Yes--> Calculate the count of metadata records + new metadata

Is Managed package metadata ? --No--> Is Custom meta data?

Is Custom meta data? --No--> Calculate the count of records + new metadata developed in X days → Save result in custom object

Is Custom meta data? --Yes--> Dynamically read custom Weightage → Calculate the count of record → Save result in custom object

Dynamically read custom Weightage → Assign Weightage to each record & then add all → Save result in custom object

Calculate the count of metadata records + new metadata → Save result in custom object

Calculate the count of metadata records + new metadata → Run Reports to show Metrics → Send report to subscribers → Stop

Calculate Configuration & Customization %

Show result on Screen → Stop

Final Metrics after calculation
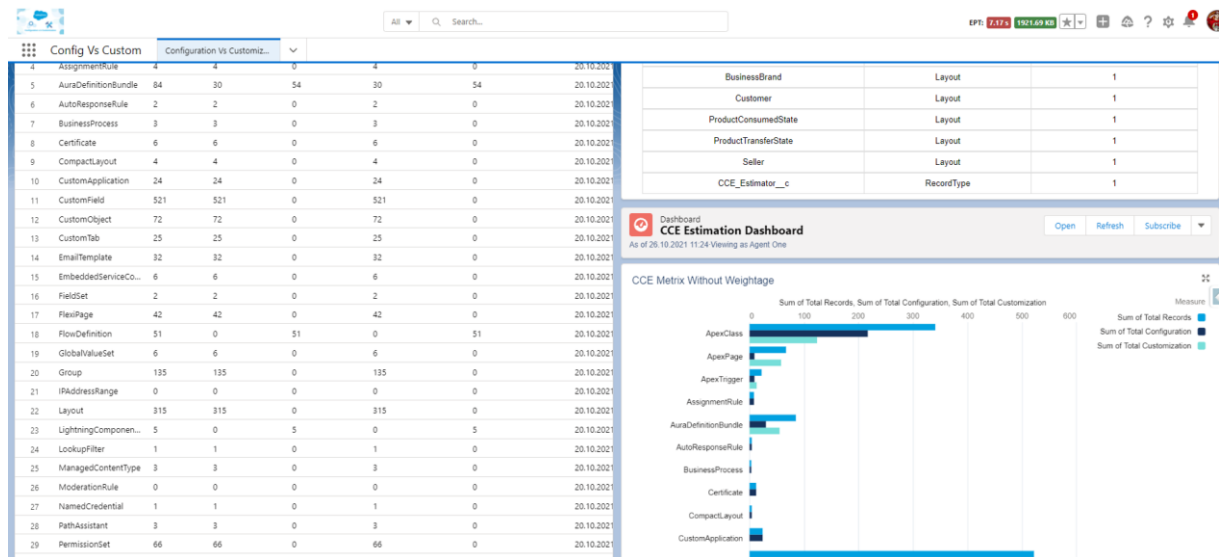
Features of Tool

Show release specific metadata Metrix

This feature enables users to see difference between previous and current release matrix. This helps users to identify how much config % and customization % development happened.

Show Object specific metadata changes per release/deployment

This feature gives an overview of metadata changes happened in a release/deployment on object(s) level. This helps users to identify specifically where exactly what type of development (custom or config) happened.

Renaming release/deployments

As this is an salesforce limitation that whenever deployment happens in salesforce, salesforce give a unique label to that deployment. This unique label is not User friendly. Hence, this by this feature user can give a specific label to each deployment/release.

Add Release Name To Release Dates

This component will help admins to give name to release dates

Select a Date

| 2023-06-07 | ▲▼ | Reset |

Release Name

| March Release | | Submit |

Escalation Feature

This is a feature of tool where admins can set Threshold limit (range) in system so that if customization and configuration exceeded beyond that threshold limit/Range then tool indicates a warning in red color. Later, admins can notify the same to responsible audience or team by sending email.

Below are the screen shots on how this feature works:

Threshold config table for Admins where admins can set the threshold limit.

CCE_Estimator_ThresholdValues

View: All Records ✓ Edit | Create New View

			New		
Action	Label ↑			Threshold Percentage	Active
Edit	Del	Configuration			1,00	✓
Edit	Del	Customization			1,00	✓

Red warning section if Threshold exceeded.

Escalate to target audiance.

Report

Reporting and Dashboard are possible in this tool. Default report is placed in console.
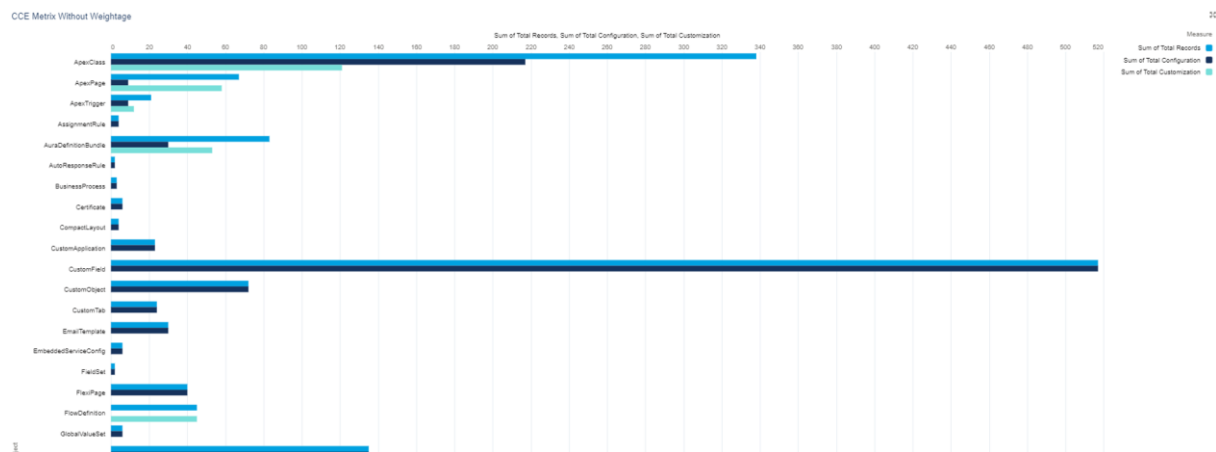
Adding new/more metadata in tool repository

This capability allows users to add more or new metadata's into tool and track them also. For that user need to navigate to "Config Vs Custom" console | click on "CCE Estimator Configuration" tab | click on New button | add metadata and its relevant details | Save.
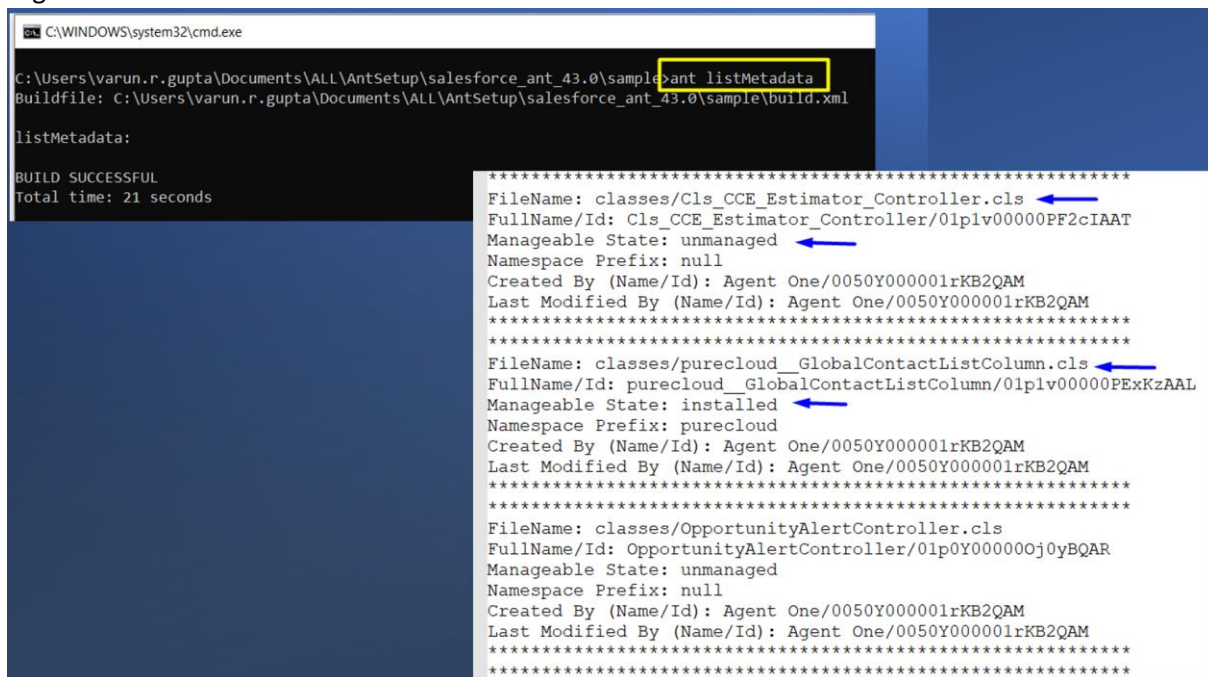
Disclamer

Same could be possible via Metadata migration tools. Ex: Ant, Copado, Auto Rabbit, etc…

Let's take example of ANT migration tool :

- Use listMetadata command. LINK

- The **listMetadata** target retrieves property information about metadata components in your organization.

Drawbacks:

- Unable to identify the size of metadata. Ex: unable to identify the line/size of Apex class and Apex trigger. **Hence, dynamically assigning customization weightage is not possible.**

- In Build.xml file we need to specify the metadata type for each metadata that we want to consider and then run command for each metadata type. Ex: one time for Apex class, then for Quick action, then for trigger, and so on…

```
<!-- Retrieve the information of all items of a particular metadata type -->
<target name="listMetadata">
  <sf:listMetadata username="${sf.username}" password="${sf.password}" sessionId="${sf.sessionId}" serverurl="${sf.serverurl}" metadataType="ApexClass"
    resultFilePath="listMetadata/list.log"/>
</target>
```

How to install tool in salesforce org

Below are the activities that need to perform to properly install tool, Up & running.

Creation of connected app

First activity is to create a connected app in system. This connected app is used internally by tool to execute tooling API internally on click of execute button.

Below are the details that need to populate in connected app:

API Name	CCE_Estimator_Connected_App
Selected OAuth Scopes	Full access (full) Perform requests at any time (refresh_token, offline_access)
Callback URL	https://<your salesforce domain>/services/authcallback/CCE_Estimator_AuthProvider

NOTE: Leave other sections as is (default)

Installation of unmanaged package

Install unmanaged package in your salesforce org.

Manual configuration

This activity is needed because via unmanaged package "Auth. Provider" & "Named Credentials" are developed in your salesforce org. But these components need your newly created connect app details.

Configure Auth. Provider

After installation of unmanaged package navigate to Auth. Provider | CCE_Estimator_AuthProvider | Replace below section details:

Consumer Key	Copy paste the Consumer Key from Connected App created above
Consumer Secret	Copy paste the Consumer Secret from Connected App created above
Authorization Endpoint URL	https://<Your salesforce domain>/services/oauth2/authorize
Token Endpoint URL	https://<Your salesforce domain>/services/oauth2/token

Save the changes.

Configure Named Credential

Navigate to Named Credentials | CCE_Estimator_NamedCredential | Replace below section details:

URL	Your salesforce domain
Authentication Provider	CCE_Estimator_AuthProvider

Save the changes.