

ASTREA ORG SCANNER



ORG SCANNER
Scan your Salesforce instance



Introduction

A huge amount of data can get stored on a Salesforce Org when the org is used over a period of time. Despite having a number of rules and strict Governance Policies, the Org can become populated with unwanted, sometimes even harmful, data. It is thus advisable to stay vigilant and keep a check on the data present in the Salesforce instance.

OrgScanner is a free and automated tool that comes handy while keeping an eye on the Org and the data contained in it. It scans the entire Org and generates a detailed pdf report based on the results of the scan. It determines the overall grade of an Org, calculated on the basis of a set of metrics. The results are broadly categorized into three general categories: Code Quality, Metadata, and Other Metrics. These are further classified into sub-categories, and the result of each sub-category is individually displayed.

Installation

OrgScanner App can be installed from the Salesforce AppExchange. One has to navigate to <https://appexchange.salesforce.com> and search this app by name, OrgScanner. During installation, if it asks for which user to grant access, we recommend selecting System Administration only.

Once the installation gets completed, you can access the OrgScanner Application from the Application drop-down list located at the top right-hand corner.

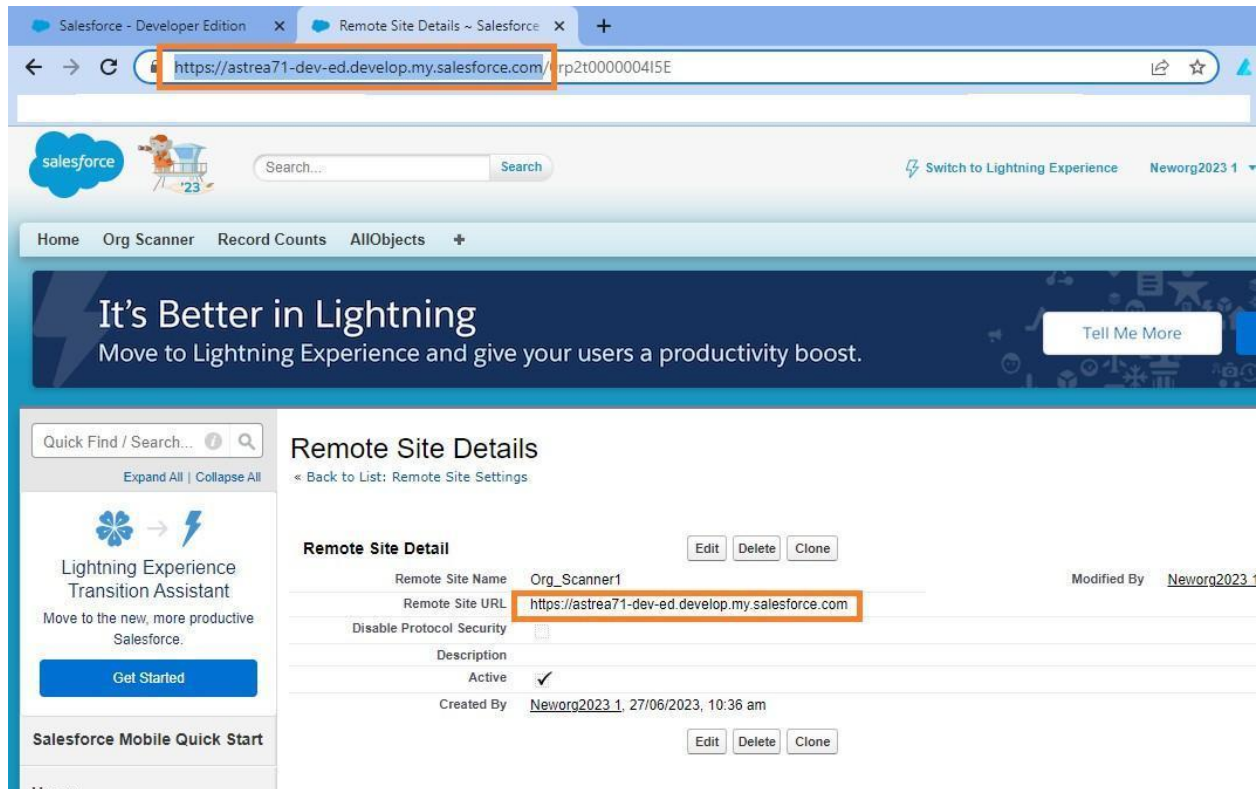
Configuration

The User needs to add two Remote site settings in their Org. Please follow the steps below:

Switch to classic salesforce Org -> Go to SetUp -> Administer -> Security Controls -> Remote Site Settings.

a) 1st site settings:

Click on the new remote site setting button -> Provide Remote site Name -> Select the org Url and place it at the remote site url field and click save. Please ensure the remote site setting should be active and added inside org.



OR

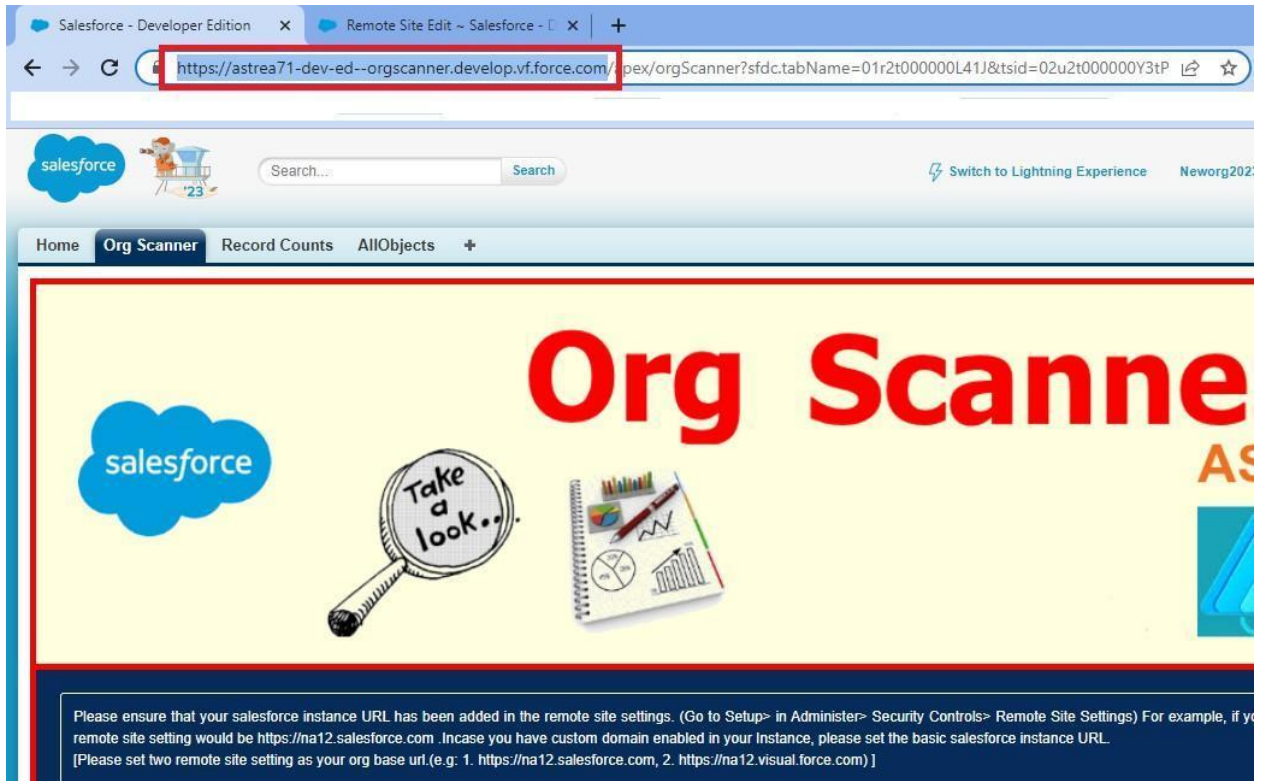
This will be your Org's base URL.

Please see below. <https://yourinstance.salesforce.com>

E.g. If your Salesforce instance is <https://na12.salesforce.com>, then the remote site setting would be <https://na12.salesforce.com> and if it is <https://cs72.salesforce.com> then it would be <https://cs72.salesforce.com>.

b) 2nd site settings:

Click on the new remote site setting button -> Provide Remote site Name -> click on the org Scanner tab and select the URL of that page and place it at the remote site url field and click save. Please ensure the remote site setting should be active and added inside the org.



OR

It will be in the below format, <https://yourinstance.visual.force.com>

E.g. If <https://na12.visual.force.com> is your salesforce instance then remote site setting would be <https://na12.visual.force.com> or if <https://cs72.visual.force.com> is the salesforce instance then it would be like this: <https://cs72.visual.force.com>.

Please ensure that your salesforce instance URL has been added in the remote site settings before proceeding any further.

Note: If you are upgrading your managed package, please add the **Code Quality Check** tab to the **Org Scanner** app.

How to Use

It is very easy to use. The User needs to select the OrgScanner app from the Application menu. Once the app opens, they need to click on the Org Scanner Tab.

Org Scanner

Home

Org Scanner

Code Quality Check



ORG SCANNER

Scan your Salesforce instance



ASTREA IT SERVICES



Please ensure that your salesforce instance URL has been added in the remote site settings. (Go to Setup> in Administer> Security Controls> Remote Site Settings) For example, if your Salesforce instance is na12.salesforce.com remote site setting would be https://na12.salesforce.com .Incase you have custom domain enabled in your Instance, please set the basic salesforce instance URL.

[Please set two remote site setting as your org base url.(e.g: 1. https://na12.salesforce.com, 2. https://na12.visual.force.com). For more help please see the User guide.]

☐ I AM A SALESFORCE ADMINISTRATOR

Please check the "I AM A SALESFORCE ADMINISTRATOR" checkbox first.

Submit

Using a Salesforce Org over time can lead to a huge amount of data being stored in the Org. Despite having a number of rules and strict Governance Policies, the Org can become populated with unwanted, sometimes even harmful, data. It is thus advisable to stay vigilant and keep a check on the data present in the Salesforce instance..

Then the user needs to check the **"I AM A SALESFORCE ADMINISTRATOR"** checkbox and click on the Submit button. Once the submit button is clicked, the user will get an email with an OrgScanner report as an attachment within a few minutes with all of the description (explained below) about the Org.

Important Note: One important thing to take note of here is that only a **Salesforce Administrator** is allowed to use this app. If a user having some other profile clicks on the Submit button then an error message (**Sorry! You cannot run this Diagnosis, only your system admin can do this.**) is displayed.

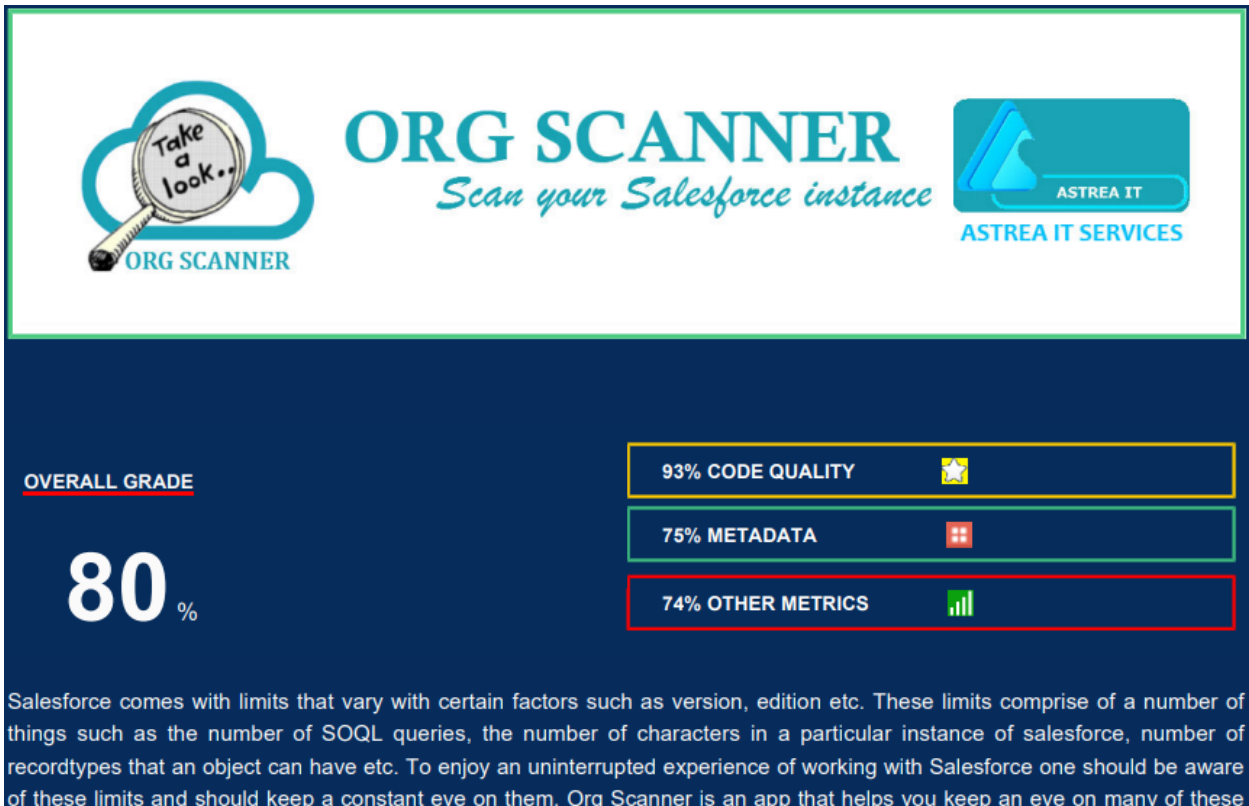
After clicking on the **Submit** button, a page will get opened displaying a success message.

Home Page

Org Diagnosis is going... The Org Scanner report is currently in progress. The Org Scanner report will be emailed to you shortly.

Thanks you so much for using the Salesforce Org Scanner. The report is in the process of being generated and should arrive to your inbox shortly.

In a short period of time, system admin will get an email with an **OrgScanner pdf** as a report which looks as shown below:



OrgScanner Report

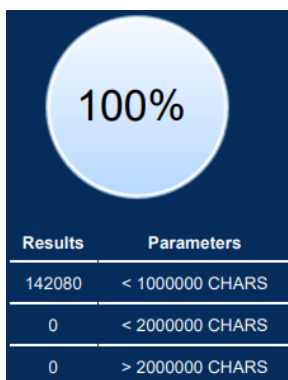
Org Scanner is an app that helps you keep an eye on many of Salesforce's limits by generating a detailed report for a Salesforce instance based on three different categories: Apex, Metadata, Others.

Apex: This category deals with Apex related features, such as Apex Triggers, Apex Classes, API Versions, Test Classes, Average Lines of Code etc.

Metadata: This category deals with the objects and their fields. It provides information about object-layout, record count, object record types, validation rules etc.

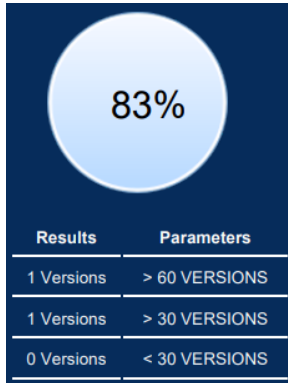
Others: This category consists of other important information of the org, like, Internal Roles, Portal Roles, Users, Permissions etc. We have these three categories in multiple modules.

1. CLASSES AND TRIGGERS: TOTAL CHARS



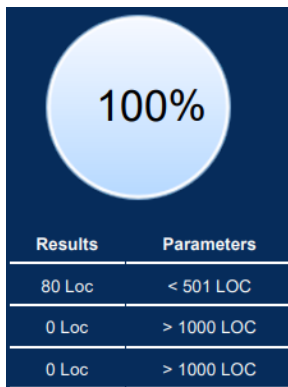
This limit defines the total number of characters that an org can have in the form of Apex Classes and Triggers. Salesforce has defined the characters limit as 3 million for every org. This part of the report counts the total number of characters in the org, returns the result in three defined base categories (less than 2000000 chars, greater than 2000000 chars, greater than 1000000 chars).

2. API VERSIONS - APEX CLASSES



The salesforce standard says that all of the apex code gets compiled and executed in the same version of salesforce Apex API. Having the classes in the recent API Version makes it easy for debugging. The code behavior and its functionality vary between different API versions. In this report the number of versions used in the org is returned in three different categories: '>30', '<30', '<60'.

3. AVG. LINE OF CODE PER CLASS



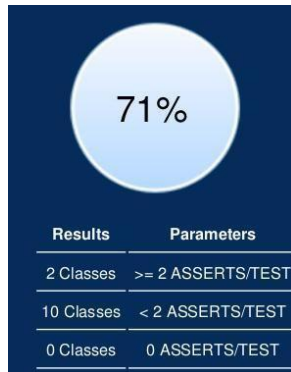
The general rule is to write a specific class in such a manner that it could achieve multiple functionalities. While doing so, it should also be kept in mind that the Lines of code should be in limit. Org scanner returns the number of classes in three different categories based on their LOCs (>501 LOC, >1000 LOC, <1000 LOC).

4. TEST CLASSES - API VERSION



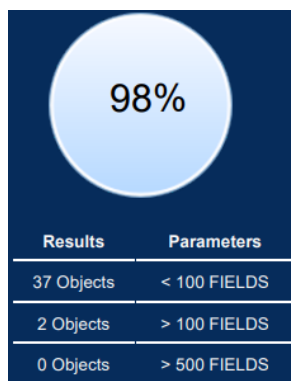
The best way of writing test classes is to insert test data in the test class itself rather than using the data from the instance. So, it should be made sure that the SeeAllData is false. The test classes need to be annotated with @isTest annotation and the best approach is to keep both, the test class and its apex class, at the same API Version. Org scanner counts the number of classes based on which category they belong to.

5. TEST CLASSES - AVG. ASSERT METHODS PER TEST METHOD



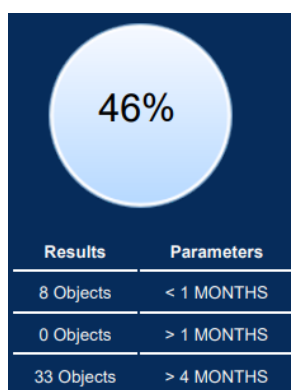
Assertions are used within test methods to make these validations. Assert methods check whether the returned results are as expected or not. Conditions are specified according to what we expect the test method to return. If the conditions are met, the code gets executed properly and if they are not, a fatal error is returned that causes code execution to halt.

6. OBJECTS - NUMBER OF FIELDS



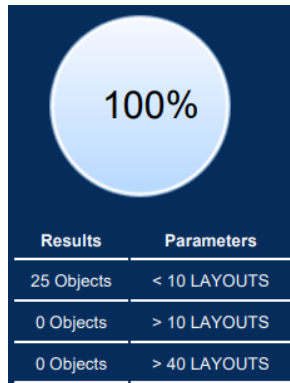
In Salesforce, Objects represent Database Tables that let us store data. Fields are attributes of Objects. All objects, whether standard or custom, have some standard fields such as name and id. In addition to these, we can create custom fields as per our requirements. However, Salesforce has set a limit on the number of fields that an object can have.

7. OBJECTS - LAST USAGE DATE



All objects have some purpose in Salesforce. If an object is not being used for quite some time, it might mean that the object is of no use and is unnecessarily occupying space in our salesforce instance. If an object is of no use then all of these things are also worthless and can consume a lot of space that can be utilized elsewhere. In such a case, the administrator can remove such an object and all of its related data from the Org.

8. OBJECTS – LAYOUTS



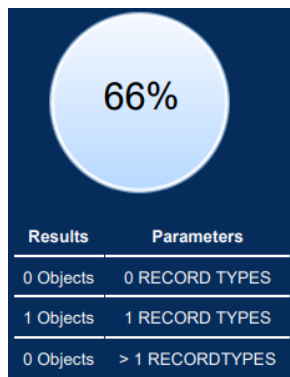
The layout of an object controls the display and arrangement of various things associated with an object (fields, buttons, related lists, etc). It is just like the decoration of a record according to the requirement of the user. Layouts in salesforce help determining the visibility of fields, whether a field is required or not, and if they are read-only or not. Different layouts for different users can be created and customized according to their specific requirements. Org Scanner tells how many layouts are available for the objects.

9. OBJECTS - RECORD COUNT



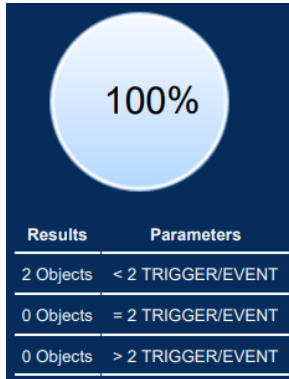
There is just one solution to optimizing the space usage: remove the unwanted data so that the Org space is freed and can be used for other important things. Record count is one such thing that tells us about the number of records in an object and is an important factor in determining the total space used in a Salesforce instance. This report returns the number of objects that have the record count of less than 10000, less than 20000, and greater than 20000.

10. OBJECTS - UNUSED RECORDTYPES



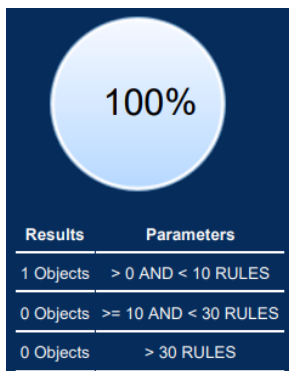
Record Types in Salesforce helps to implement our custom business processes. They allow us to offer different business processes, picklist values, and page layouts to different users based on their specific requirements. However, sometimes there are record types that are not used and provide no value, thus increasing the administrative burden. Such unused record types should be removed, as and when possible.

11. OBJECTS - TRIGGERS ON THE SAME EVENT



There are several best practices associated with trigger development, one of them being: “For an object, there should be only one trigger for one event”. Having more than one trigger on the same event for one object can lead to unexpected results. Thus, it is important to keep track of triggers on the same event. Org Scanner tells about how many objects have triggers on the same event.

12. OBJECTS - VALIDATION RULE ACTIVE



Validation Rules are like borders that prevent the enemy from entering. They regulate the entry of unwanted data by defining various conditions. However, having a lot of validation rules can lead to an increase in the difficulty in entering data and would also increase the complexity for the org administrator to keep an eye on all of them.

13. OBJECTS - VALIDATION RULE INACTIVE



As mentioned earlier, Validation rules are of utmost importance in any Salesforce instance but if a Validation Rule is inactive, it has no value and is rendered useless. An inactive Validation Rule won't prevent invalid data from being saved in the org and would occupy space. Such rules can be removed from the org to make space for new and more important things. Before removing, their backup can be taken for future instance using version control repository using the metadata API.

14. OBJECTS - WORKFLOW RULE ACTIVE



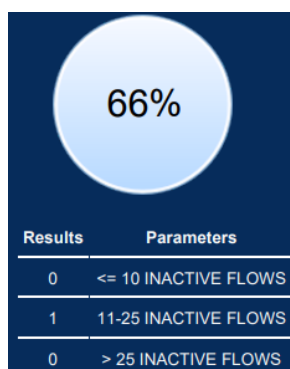
Just like Validation Rules, Workflows play an important role in an org. They should also be specific in number to reduce the complexity and understandability. A lot of workflow rules on the same object can create confusion in execution and may also introduce bugs thus leading to unexpected results and exceptions. The number of active workflow rules should thus be regulated from time to time.

15. OBJECTS - WORKFLOW RULE INACTIVE



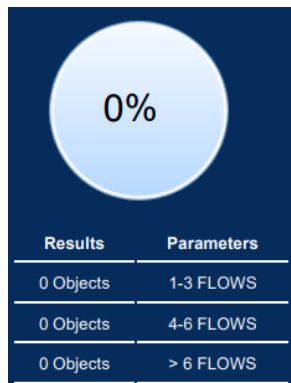
Sometimes, to fulfill the requirements, the need arises to create workflows but later they are made inactive as the requirement changes. Workflow rules which are inactive are of no use to the Salesforce org. They are worthless and consume space in the org. They should be removed to simplify the org, thus making it easier to maintain.

16. OBJECTS - FLOW STATUS INACTIVE



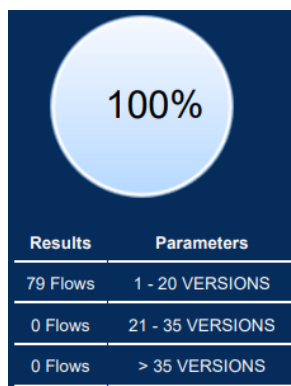
Salesforce stores inactive flow definitions when flows are deactivated or replaced with newer versions. A large number of inactive flows can make the org harder to maintain and may indicate that obsolete automation should be reviewed and cleaned up periodically.

17. OBJECTS - RECORD TRIGGERED FLOWS PER OBJECT



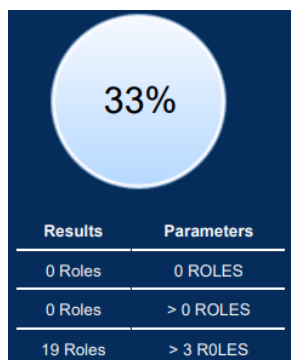
Record-triggered flows automate business processes when records are created, updated, or deleted. Having too many record-triggered flows on the same object can increase automation complexity, make debugging difficult, and negatively impact performance. This metric identifies objects with a high number of record-triggered flows so unnecessary automation can be reviewed and consolidated.

18. OBJECTS - FLOW VERSION



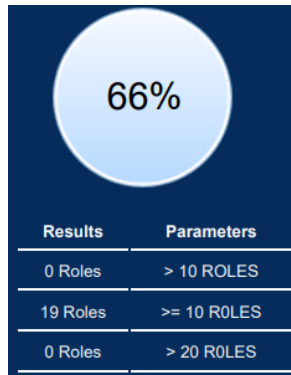
Every time a Flow is modified and activated, Salesforce creates a new version. As the version number increases, the Flow approaches Salesforce's maximum supported limit of 50 versions. Flows with a high version count should be reviewed and cleaned up to avoid deployment and maintenance issues.

19. INTERNAL ROLES - UNASSIGNED



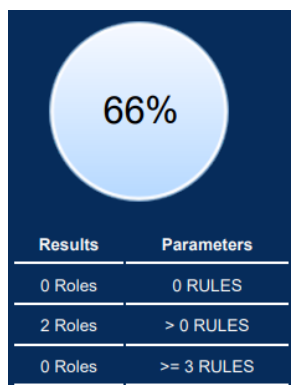
Salesforce offers User Role Hierarchy that can be used with sharing settings to determine the levels of access that users can have to one's Salesforce org's data. Roles control a user's record-level access permissions through role hierarchy and sharing rules. Internal Roles that are of no use can be removed from the org so that the overhead of the org is reduced.

20. INTERNAL ROLES – TOTAL



Hierarchy makes things simple for understanding, but when security and performance come into concern then the hierarchies do not tend to be supporting enough. Similar is the case with Role hierarchy. When a large number of roles are added, it increases the complexity of the org and if the roles have higher levels of hierarchy then the performance of the org is slowed down because it takes a lot of time to solve the complexity at each level of the hierarchy.

21. PORTAL ROLES – UNASSIGNED



Salesforce's limit on the number of portal roles that can be used in an org's communities and portals is 5000. To prevent an added overhead in this number, the portal roles should be reviewed timely and the unassigned roles and roles assigned to inactive users should be deleted. This reduces the complexity and thus improves the overall performance of the salesforce instance.

22. PORTAL ROLES - TOTAL



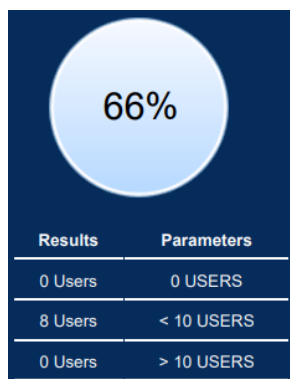
A large number of portal roles adds to the complexity and impacts the Org performance. As the number of Portal Roles increases, the hierarchy level in the org increases. Also, there will be more chances of having a lot of unused Portal Roles that will occupy an unnecessary space in the org. Thus, make sure to clear unused roles or roles for inactive portal users.

23. USER - ADMIN USERS (MODIFY ALL DATA)



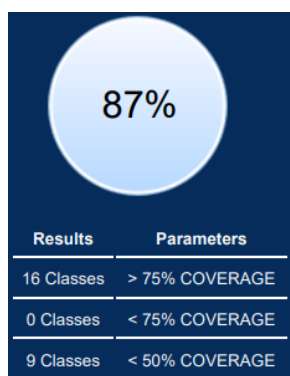
Users with 'Modify All Data' permission can modify all apps and data, even if the apps and data are not shared with them. They have permission for data cleansing, deduplication, mass deletion, mass transferring, and managing record approvals. In this part of the report, Org Scanner returns the number of Users that have Admin User's rights.

24. USER - NOT LOGGED IN LAST 30 DAYS



There might be users who have not logged in for quite some time for several reasons. In some cases, this can be because the user left the company. In such a case, it is possible that we won't need an extra user in our Salesforce instance. Hence, it is important to keep track of Users who are not very active. Org Scanner returns the number of such users who have not logged in for the last 30 days.

25. COVERAGE - CLASSES AND TRIGGERS COUNT IN SPECIFIED RANGE



In Salesforce, it is mandatory to have a minimum of 75% test coverage for Classes and Triggers to make them available for deployment. While developing Apex Classes and Triggers we need to make sure that they meet this criterion. This last part of the generated report deals with this. It tells us about how many classes and triggers have 75% above coverage, how many have coverage less than 75% and how many of them fall in the category of below 50%.

Percentage Calculation

All the results are returned based on three different conditional parameters (greater than something, less than the other thing and so on) and the final percentage is calculated. The criterion for percent calculation is as follows:

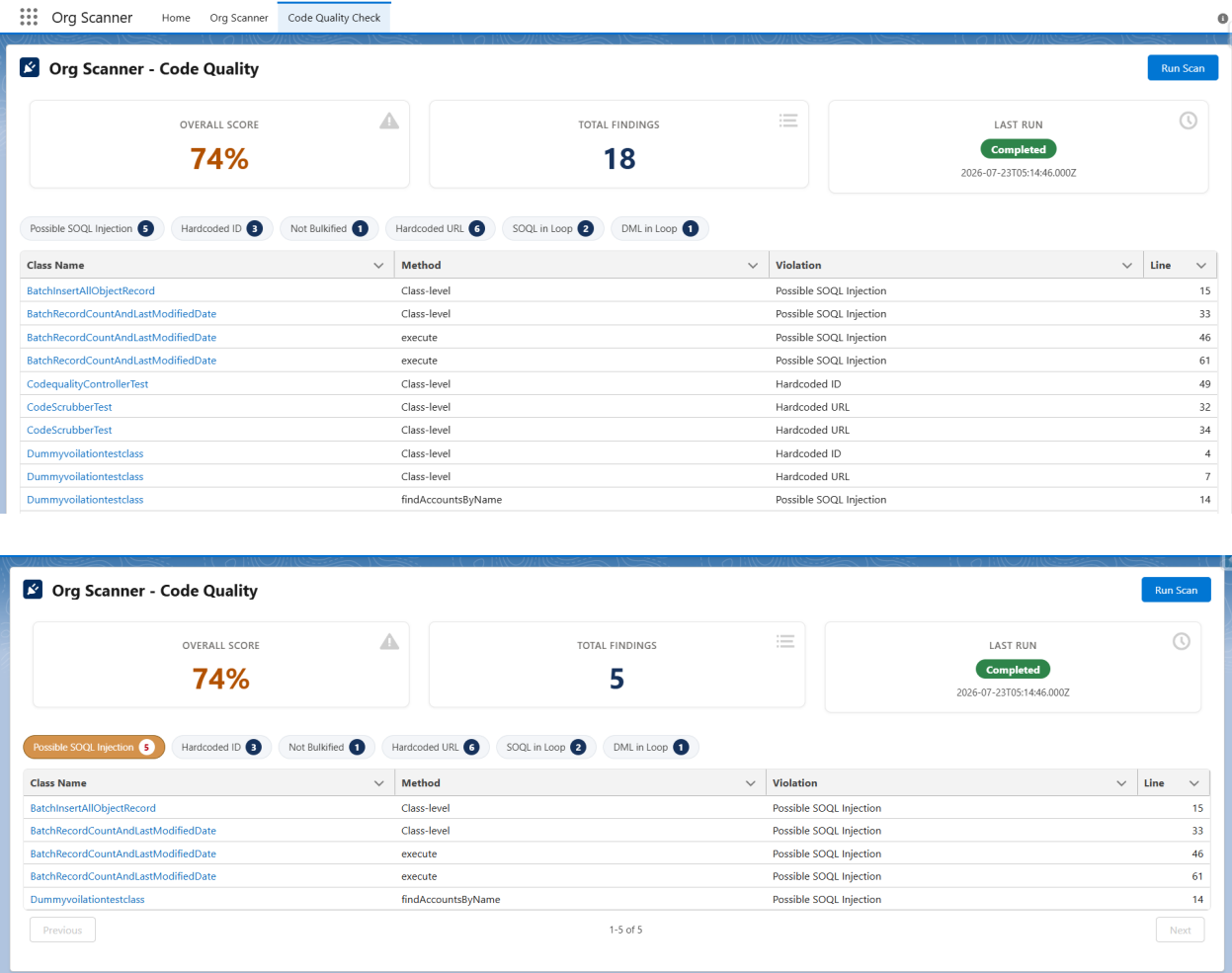
Result	Parameters	Multiplier
Result 1	Condition 1	100
Result 2	Condition 2	66
Result 3	Condition 3	33

$\% = [(result\ 1 * 100) + (result\ 2 * 66) + (result\ 3 * 33)] / (result\ 1 + result\ 2 + result\ 3)$

Note: For any assistance you may reach us on: ngabrani@astreait.com

Code Quality Check

Note: If you are upgrading your managed package, please add the **Code Quality Check** tab to the **Org Scanner** app.



Overview

The **Code Quality Check** feature analyzes the Apex code in your Salesforce Org and identifies common coding issues based on Salesforce best practices. It helps administrators and developers quickly locate potential security, performance, and maintainability problems without manually reviewing every Apex class.

The scan examines Apex Classes and Triggers and generates a report containing the overall code quality score, total findings, and detailed information about each detected violation.

How to Use

- a) Open the Code Quality Check tab.
- b) Click the Run Scan button.
- c) Wait for the scan to complete.
- d) Review the generated summary and the list of detected violations.

Dashboard Information

Overall Score

Displays the overall code quality percentage of the scanned Apex code.

A higher percentage indicates better adherence to Salesforce coding best practices.

Total Findings

Shows the total number of violations detected during the scan.

Last Run

Displays:

- Scan Status
- Date and Time of the last completed scan

Violation Summary

The dashboard categorizes detected violations for quick analysis. Each category displays the total number of occurrences found during the scan. Selecting a violation category filters the results and displays only the Apex Classes and methods affected by that specific issue, allowing developers to quickly identify and navigate to the relevant code for remediation.

Detailed Violation Report

The detailed results section lists every violation detected during the scan. For each issue, it displays the **Apex Class** in which the violation was found, the specific **method** containing the issue, the type of **violation** detected, and the exact **line number** where it occurs. This enables developers to quickly locate the affected code, understand the issue, and resolve it efficiently without manually searching through the source code.

Supported Violations

Possible SOQL Injection

Detects dynamic SOQL queries that are not properly sanitized and may expose the application to SOQL Injection attacks.

Hardcoded ID

Detects Salesforce Record IDs written directly in Apex code. Hardcoded IDs reduce portability between environments and should be replaced with dynamic queries or Custom Metadata.

Hardcoded URL

Identifies URLs embedded directly in Apex code. URLs should be configurable using Custom Metadata, Custom Settings, or Named Credentials.

SOQL in Loop

Detects SOQL queries executed inside loops. This can easily exceed Salesforce Governor Limits and impact performance.

DML in Loop

Detects DML operations performed inside loops. Bulkifying DML operations improves performance and prevents governor limit exceptions.

Not Bulkified

Detects code that is not designed to process multiple records efficiently. Bulkified code is required to support triggers, batch jobs, and large data volumes.

Benefits

- Improves code quality.
- Detects security vulnerabilities.
- Helps prevent Governor Limit exceptions.
- Simplifies code reviews.
- Makes Apex code more scalable and maintainable.
- Provides the exact class, method, and line number for faster issue resolution.

Note: For any assistance you may reach us on: ngabrani@astreait.com