



A Marketing Automation Corp Company

User Guide

Status Tracker for Salesforce

Introduction

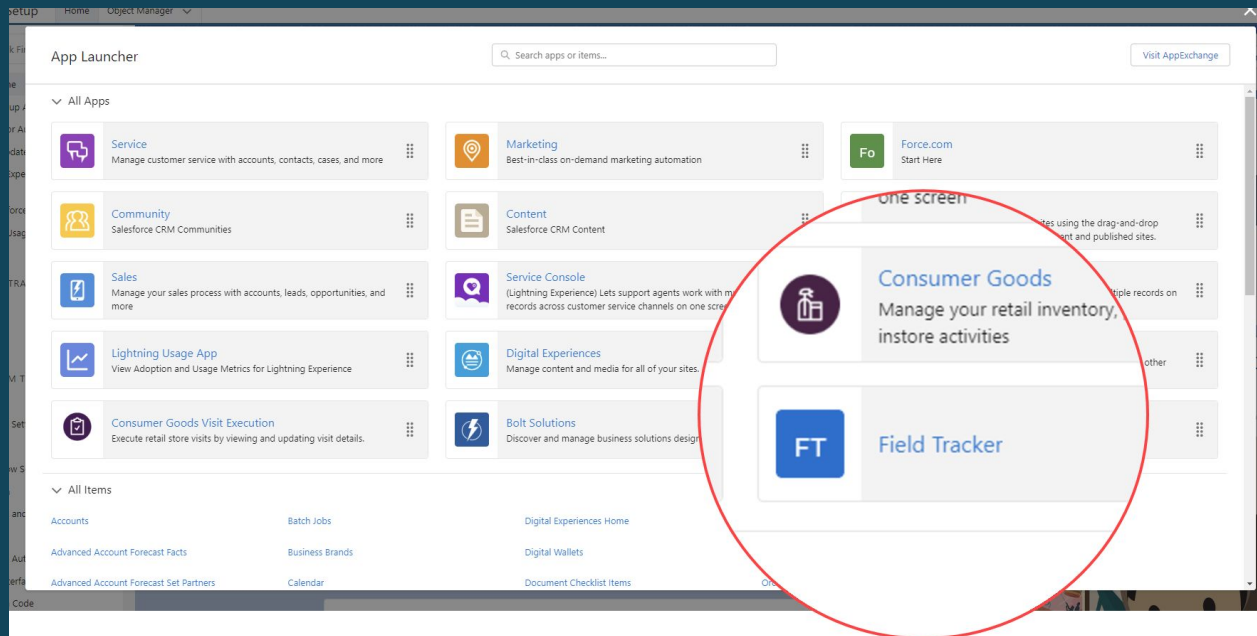
Status tracker for Salesforce has the ability to track history for any field on any object! Most standard objects are supported and all custom objects are supported

Status tracker for Salesforce is a 100% Salesforce native application. The admins can track the history of any changes that are made in objects by showing the previous, as well as, the updated value. The admins can also dynamically configure any object.

To track the changes that happened in the field, all you need to do is select a certain field to track and display the field history in the history related list of an object.

Enable Tracking for Standard & Custom Objects

In App Launcher find “Field Tracker”



Enable Tracking for Standard & Custom Objects

Click on “Quick Links → Manage Tracking Settings”

 Field Tracker

Welcome

Quick Links

MARKETING
AUTOMATION
CANADA

Quick Links

Get User Guide

Download

Manage Tracking Settings

Manage Tracking Setting

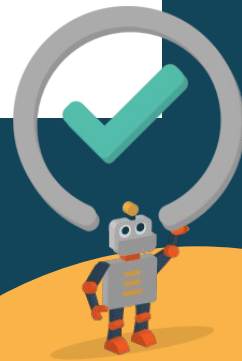
User can create a Status Tracking Setting to enable field tracking for specific object fields.
Please refer below feilds:

- **Object API** : API name of object to be tracked.
- **Picklist Field API to Be Tracked** API name of Picklist field to be tracked from above object.
- **Lookup Field API** : Lookup Field API
- **Is Active** : This field used to activate the configuration and deactivate it.

Status Tracking Object Defination

Manage Status Tracking Object

- This is the object to store tracking infomration.
- All tracking related fields can be set from here.
- User can create lookups to the object, which needs to be tracked.
- User can define specific fields to display on the layout.



Configuration Status Tracking Settings

User can create a Status Tracking Setting to enable field tracking for specific object fields. Please refer the following field definitions for configuration:

Field Name	Description
Object API	API name of object to be tracked.
Picklist Field API to Be Tracked	API name of Picklist field to be tracked from above object.
Lookup Field API	Lookup field on Status Tracking object which is used to link the tracking record with actual record.
Is Active	This field used to activate the configuration and deactivate it.

 **SETUP**

Custom Metadata Types

Status Tracking Settings

View: All Create New View

Action

Label ↑

Edit | Del

Lead_Status

Edit | Del

Opp_Field

Edit | Del

Opp_Demo_Prep

Edit | Del

Product Demo Tracking

Edit | Del

Random Picklist Field_c_tracking

New

Status Tracking Setting Name

Lead_Status

Opp_Field

Opp_Demo_Prep

Product Demo Tracking

Random_Picklist_Field_c_tracking

Namespace Prefix

MACCRM

MACCRM

MACCRM

MACCRM

MACCRM

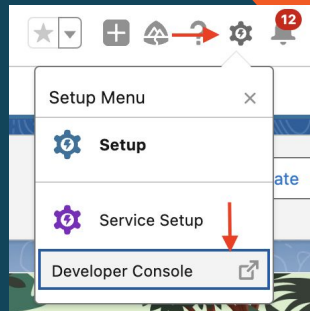


Configuration of the Trigger

Adding a Trigger Snippet in Dev Console

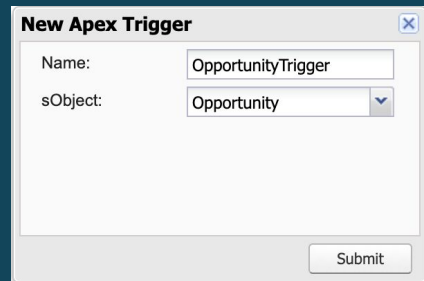
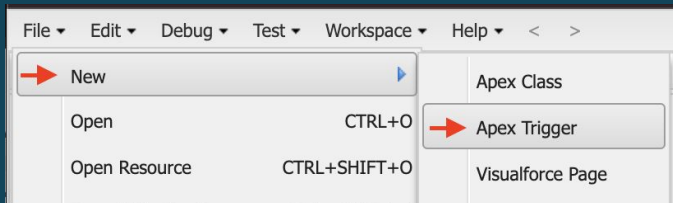
1. Navigate to the Developer Console:

- Open Salesforce and click on your avatar or name in the upper-right corner.
- Select Developer Console.



2. Create an Apex Trigger:

- Go to File → New → Apex Trigger.
- Name your trigger, e.g., OpportunityTrigger.
- Select the object, e.g., Opportunity, and click Submit.

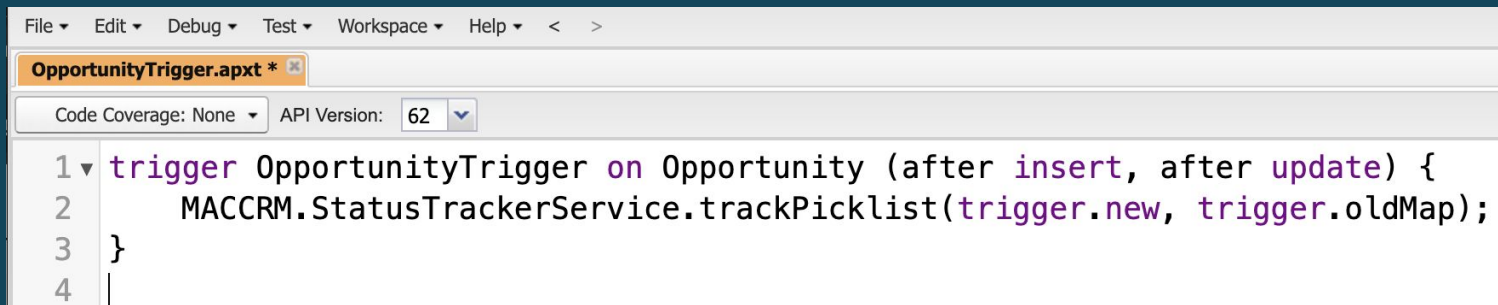


Configuration of the Trigger

Paste the following snippet code:

```
trigger OpportunityTrigger on Opportunity (after insert, after update) {  
    MACCRM.StatusTrackerService.trackPicklist(trigger.new, trigger.oldMap);  
}
```

Save the Trigger: Ensure there are no syntax errors and save your work.



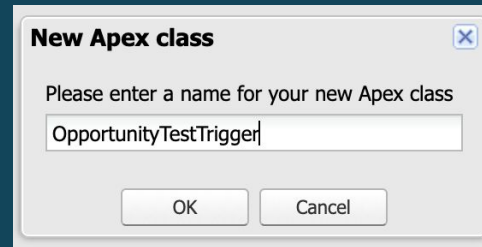
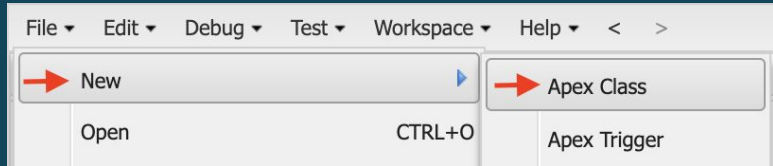
Developing Test Class

Creating a Test Class for the Trigger

1. Go to Developer Console:

- In the Developer Console, click File → New → Apex Class.
- Name the test class, e.g., OpportunityTriggerTest.

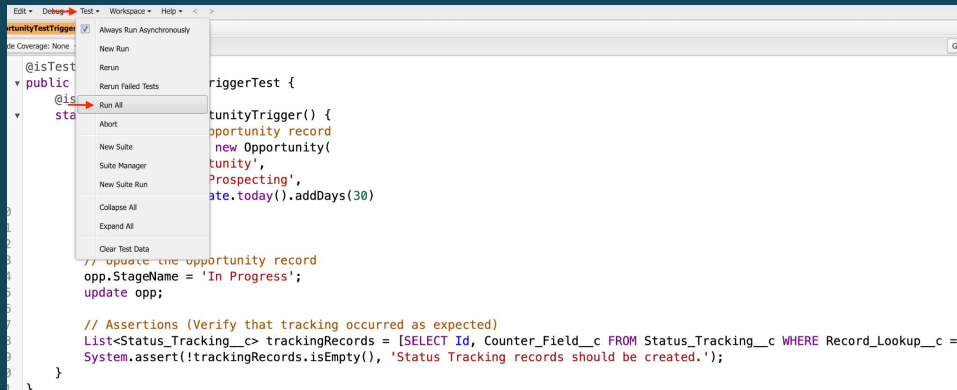
2. Develop the Test Class



Developing Test Class

Paste the following snippet code:

```
@isTest
public class OpportunityTriggerTest {
    @isTest
    static void testOpportunityTrigger() {
        // Create a new Opportunity record
        Opportunity opp = new Opportunity(
            Name = 'Test Opportunity',
            StageName = 'Open',
            CloseDate = Date.today().addDays(30)
        );
        insert opp;
        // Update the Opportunity record
        opp.StageName = 'In Progress';
        update opp;
        // Assertions (Verify that tracking occurred as expected)
        List<Status_Tracking__c> trackingRecords = [SELECT Id, Counter_Field__c FROM Status_Tracking__c
        WHERE Record_Lookup__c = :opp.Id];
        System.assert(!trackingRecords.isEmpty(), 'Status Tracking records should be created.');
```



Note: StageName and opp.StageName variables can vary depending on your instance

Developing Test Class

1. Run the Test Class:

- In the Developer Console, navigate to Test → New Run.
- Select the OpportunityTriggerTest class and execute the test.

2. Check the Results:

- Ensure the test passes and generates appropriate tracking records.

3. Verify Status:

- Ensure your trigger is active and error-free.
- Test coverage should meet Salesforce requirements (minimum 75%).

Note:

- Test class is mandatory to ensure functionality
- If test class is already present in your system which covers a stage update scenario, no additional test class is required.
- This test class is a general test class, if your org has some mandatory validations, please ensure the test class is accordingly written.

Associate the Related List to the Record Page

To add the Status Tracking Related List to a record page, please refer the following:

1. Navigate to Setup:

- Go to Setup → Lightning App Builder.
- Open the Record Page for the object you are configuring (e.g., Opportunity).

2. Edit the Page Layout:

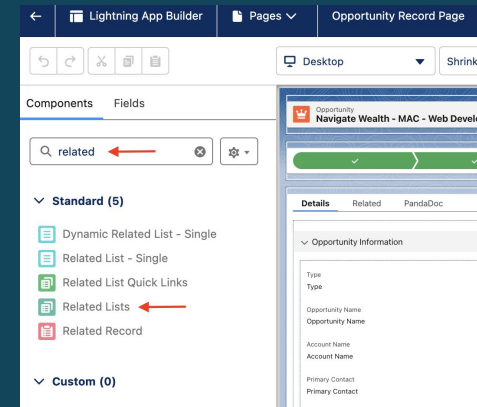
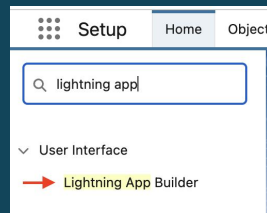
- Click Edit to open the Lightning App Builder.

3. Add the Related List:

- Drag the Related List - Single standard component onto the page layout.
- In the component settings, set the related object name to Status Tracking.

4. Save and Activate:

- Save the changes and activate the updated record page.



Reports on the Status Tracking

User can use the “Status Tracking” Object to generate reports for the field history. Please refer the following table for field definitions:

Field Name	Description
Record Lookup	This is a lookup field of respective objects like Account, Contact or any custom object the user configures.
Counter Days	Number of days specific value is present for tracking field.
Counter Field	Field name that's being tracked.
Counter Value	Value of tracking field.
From Date	Date that specific value is set for Counter Field.
To Date	Date that the field value is changed.
User	The user who updated the field value.



A Marketing Automation Corp Company

Thank you!

crm-forge.com