

Apex Triggers

1. Build Apex Triggers - *ClosedOpportunityTrigger*

```
trigger ClosedOpportunityTrigger on Opportunity (after insert, after update) {  
    List<task> tasklist = new List<Task>();  
  
    for(Opportunity opp: Trigger.New){  
        if(opp.StageName == 'Closed Won'){  
            tasklist.add(new Task(Subject = 'Follow Up Test Task', WhatId = opp.Id));  
        }  
    }  
  
    if(tasklist.size()>0){  
        insert tasklist;  
    }  
}
```

Asynchronous Apex

Use Future Methods--AccountProcessor

```
public class AccountProcessor {  
  
    @Future  
    public static void countContacts(List<Id> accountIds){  
        Map<Id,List<Contact>> accContacts = new Map<Id,List<Contact>>();  
        List<Account> accsForUpdate = new List<Account>();  
        if(accountIds != null){  
            For(Account acc : [SELECT id,(SELECT id,Name FROM Contacts)FROM Account  
            where id in: accountIds]){  
                accContacts.put(acc.Id,acc.contacts);  
            }  
        }  
    }  
}
```

```

        for(Id key : accContacts.keySet()){
            Account a = New Account(id = key);
            a.Number_of_Contacts__c = accContacts.get(key).size();
            accsForUpdate.add(a);
        }
        update accsForUpdate;
    }
}
}

```

AccountProcessor Test

```

@isTest
public class AccountProcessorTest {

    @testSetup
    static void setupAccount(){

        List<Account> accounts =
        RandomAccountContactFactory.generateRandomAccounts(1);
        insert accounts;

        List<Contact> contacts =
        RandomAccountContactFactory.generateRandomContacts(3, 'TestAP' ,
        accounts.get(0).id);
        insert contacts;

    }

    @isTest
    static void testAccountProcessor(){

        List<id> accIds = new List<id>();
        for(Account a : [select id from Account]){
            accIds.add(a.id);
        }
    }
}

```

```
    Test.startTest();
    AccountProcessor.countContacts(acctIds);
    Test.stopTest();

}

}
```

Use Batch Apex--LeadProcessor

```
global class LeadProcessor implements
Database.Batchable<sObject>, Database.Stateful {

    // instance member to retain state across transactions
    global Integer recordsProcessed = 0;

    global Database.QueryLocator start(Database.BatchableContext bc) {
        return Database.getQueryLocator('SELECT Id, LeadSource FROM Lead');
    }

    global void execute(Database.BatchableContext bc, List<Lead> scope){
        // process each batch of records
        List<Lead> leads = new List<Lead>();
        for (Lead lead : scope) {

            lead.LeadSource = 'Dreamforce';
            // increment the instance member counter
            recordsProcessed = recordsProcessed + 1;

        }
        update leads;
    }

    global void finish(Database.BatchableContext bc){
        System.debug(recordsProcessed + ' records processed. Shazam!');
    }
}
```

```
}
```

LeadProcessor Test

```
@isTest
public class LeadProcessorTest {
    @testSetup
    static void setup() {
        List<Lead> leads = new List<Lead>();
        // insert 200 leads
        for (Integer i=0;i<200;i++) {
            leads.add(new Lead(LastName='Lead '+i,
                Company='Lead', Status='Open - Not Contacted'));
        }
        insert leads;
    }

    static testmethod void test() {
        Test.startTest();
        LeadProcessor lp = new LeadProcessor();
        Id batchId = Database.executeBatch(lp, 200);
        Test.stopTest();

        // after the testing stops, assert records were updated properly
        System.assertEquals(200, [select count() from lead where LeadSource =
'Dreamforce']);
    }
}
```

Control Process with Queueable Apex --- AddPrimaryContact

```
public class AddPrimaryContact implements Queueable {
    public contact c;
    public String state;

    public AddPrimaryContact(Contact c, String state) {
```

```

        this.c = c;
        this.state = state;
    }

    public void execute(QueueableContext qc) {
        system.debug('this.c = '+this.c+' this.state = '+this.state);
        List<Account> acc_lst = new List<account>([select id, name, BillingState from account where account.BillingState = :this.state limit 200]);
        List<contact> c_lst = new List<contact>();
        for(account a: acc_lst) {
            contact c = new contact();
            c = this.c.clone(false, false, false, false);
            c.AccountId = a.Id;
            c_lst.add(c);
        }
        insert c_lst;
    }

}

```

AddPrimaryContact Test

```

@IsTest
public class AddPrimaryContactTest {

    @IsTest
    public static void testing() {
        List<account> acc_lst = new List<account>();
        for (Integer i=0; i<50;i++) {
            account a = new account(name=string.valueOf(i),billingstate='NY');
            system.debug('account a = '+a);
            acc_lst.add(a);
        }
        for (Integer i=0; i<50;i++) {
            account a = new account(name=string.valueOf(50+i),billingstate='CA');
            system.debug('account a = '+a);
            acc_lst.add(a);
        }
    }
}

```

```

        }
        insert acc_lst;
        Test.startTest();
        contact c = new contact(lastname='alex');
        AddPrimaryContact apc = new AddPrimaryContact(c,'CA');
        system.debug('apc = '+apc);
        System.enqueueJob(apc);
        Test.stopTest();
        List<contact> c_lst = new List<contact>([select id from contact]);
        Integer size = c_lst.size();
        system.assertEquals(50, size);
    }

}

```

Schedule Jobs Using The Apex Scheduler Unit--DailyLeadProcessor

```

global class DailyLeadProcessor implements Schedulable {

    global void execute(SchedulableContext ctx) {
        List<Lead> lList = [Select Id, LeadSource from Lead where LeadSource = null];

        if(!lList.isEmpty()) {
            for(Lead l: lList) {
                l.LeadSource = 'Dreamforce';
            }
            update lList;
        }
    }
}

```

DailyLeadProcessorTest

```

isTest
private class DailyLeadProcessorTest {

```

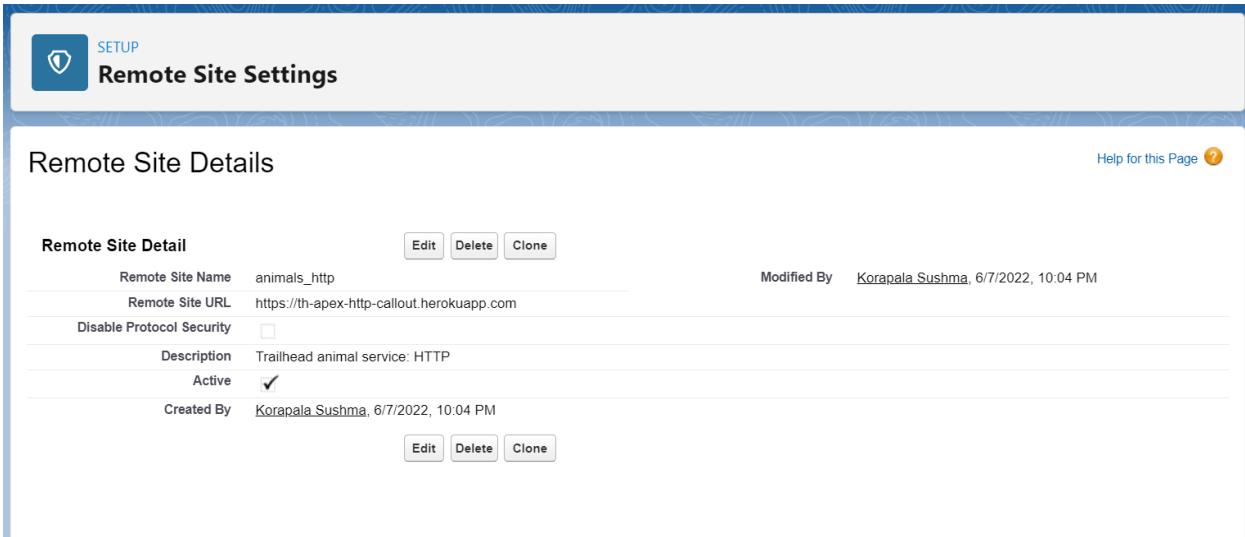
```

static testMethod void testDailyLeadProcessor() {
    String CRON_EXP = '0 0 1 * * ?';
    List<Lead> IList = new List<Lead>();
    for (Integer i = 0; i < 200; i++) {
        IList.add(new Lead(LastName='Dreamforce'+i, Company='Test1
Inc.', Status='Open - Not Contacted'));
    }
    insert IList;

    Test.startTest();
    String jobId = System.schedule('DailyLeadProcessor', CRON_EXP, new
DailyLeadProcessor());
}

```

Apex Integration Services-- Apex Integration Overview



The screenshot shows the 'Remote Site Settings' page in the Salesforce Setup. A single remote site named 'animals_http' is listed. The details are as follows:

Remote Site Detail		Edit	Delete	Clone
Remote Site Name	animals_http			
Remote Site URL	https://th-apex-http-callout.herokuapp.com			
Disable Protocol Security	<input type="checkbox"/>			
Description	Trailhead animal service: HTTP			
Active	<input checked="" type="checkbox"/>			
Created By	Korapala Sushma, 6/7/2022, 10:04 PM			
		Edit	Delete	Clone

Modified By: Korapala Sushma, 6/7/2022, 10:04 PM

Help for this Page [?](#)

The screenshot shows the 'Remote Site Settings' page in the Salesforce Setup. The top navigation bar includes 'SETUP' and the 'Remote Site Settings' section. Below the header, the title 'Remote Site Details' is displayed, along with a 'Help for this Page' link.

Remote Site Detail

		Edit	Delete	Clone
Remote Site Name	animals_soap			
Remote Site URL	https://th-apex-soap-service.herokuapp.com			
Disable Protocol Security	<input type="checkbox"/>			
Description	Trailhead animal service: SOAP			
Active	<input checked="" type="checkbox"/>			
Created By	Korapala Sushma, 6/7/2022, 10:04 PM			
Edit Delete Clone				

Modified By [Korapala Sushma](#), 6/7/2022, 10:04 PM

Apex REST Callouts--AnimalLocator

```
public class AnimalLocator {
    public class cls_animal {
        public Integer id;
        public String name;
        public String eats;
        public String says;
    }
    public class JSONOutput{
        public cls_animal animal;
        //public JSONOutput parse(String json){
        //return (JSONOutput) System.JSON.deserialize(json, JSONOutput.class);
        //}
    }
}
```

```

public static String getAnimalNameById (Integer id) {
    Http http = new Http();
    HttpRequest request = new HttpRequest();
    request.setEndpoint('https://th-apex-http-callout.herokuapp.com/animals/' + id);
    //request.setHeader('id', String.valueOf(id)); -- cannot be used in this challenge :)
    request.setMethod('GET');
    HttpResponse response = http.send(request);
    system.debug('response: ' + response.getBody());
    //Map<String, Object> map_results = (Map<String, Object>)
    JSON.deserializeUntyped(response.getBody());
    jsonOutput results = (jsonOutput) JSON.deserialize(response.getBody(), jsonOutput.class);
    //Object results = (Object) map_results.get('animal');

    system.debug('results= ' + results.animal.name);
    return(results.animal.name);
}

}

```

AnimalLocator Test

```

@IsTest
public class AnimalLocatorTest {
    @isTest
    public static void testAnimalLocator() {
        Test.setMock(HttpCalloutMock.class, new AnimalLocatorMock());
        //Httpresponse response = AnimalLocator.getAnimalNameById(1);
        String s = AnimalLocator.getAnimalNameById(1);
        system.debug('string returned: ' + s);
    }
}

```

AnimalLocatorMock

```

@IsTest
global class AnimalLocatorMock implements HttpCalloutMock {

    global HTTPResponse respond(HTTPRequest request) {

```

```

Httpresponse response = new Httpresponse();
response.setStatusCode(200);
//-- directly output the JSON, instead of creating a logic
//response.setHeader('key, value')
//Integer id = Integer.valueOf(request.getHeader('id'));
//Integer id = 1;
//List<String> lst_body = new List<String> {'majestic badger', 'fluffy bunny'};
//system.debug('animal return value: ' + lst_body[id]);
response.setBody('{"animal":{"id":1,"name":"chicken","eats":"chicken food","says":"cluck
cluck"}}');
return response;
}

}

```

Apex SOAP callouts--ParkService

```

public class ParkService {
    public class byCountryResponse {
        public String[] return_x;
        private String[] return_x_type_info = new String[]{'return','http://parks.services/','null','0'-
        '1','false'};
        private String[] apex_schema_type_info = new String[]{'http://parks.services/','false','false'};
        private String[] field_order_type_info = new String[]{'return_x'};
    }
    public class byCountry {
        public String arg0;
        private String[] arg0_type_info = new String[]{'arg0','http://parks.services/','null','0','1','false'};
        private String[] apex_schema_type_info = new String[]{'http://parks.services/','false','false'};
        private String[] field_order_type_info = new String[]{'arg0'};
    }
    public class ParksImplPort {
        public String endpoint_x = 'https://th-apex-soap-service.herokuapp.com/service/parks';
        public Map<String,String> inputHttpHeaders_x;
        public Map<String,String> outputHttpHeaders_x;
        public String clientCertName_x;
        public String clientCert_x;
        public String clientCertPasswd_x;
        public Integer timeout_x;
        private String[] ns_map_type_info = new String[]{'http://parks.services/','ParkService'};
    }
}

```

```

public String[] byCountry(String arg0) {
    ParkService.byCountry request_x = new ParkService.byCountry();
    request_x.arg0 = arg0;
    ParkService.byCountryResponse response_x;
    Map<String, ParkService.byCountryResponse> response_map_x = new Map<String,
ParkService.byCountryResponse>();
    response_map_x.put('response_x', response_x);
    WebServiceCallout.invoke(
        this,
        request_x,
        response_map_x,
        new String[]{endpoint_x,
        '',
        'http://parks.services/',
        'byCountry',
        'http://parks.services/',
        'byCountryResponse',
        'ParkService.byCountryResponse'}
    );
    response_x = response_map_x.get('response_x');
    return response_x.return_x;
}
}
}
}

```

ParkServiceMock

```

@isTest
global class ParkServiceMock implements WebServiceMock {
    global void doInvoke(
        Object stub,
        Object request,
        Map<String, Object> response,
        String endpoint,
        String soapAction,
        String requestName,
        String responseNS,
        String responseName,
        String responseType) {
    ParkService.byCountryResponse response_x =
        new ParkService.byCountryResponse();

```

```

        response_x.return_x = new List < String > {'a', 'b'};
        response.put('response_x', response_x);
    }
}

```

ParkLocator

```

public class ParkLocator {
    public static List < String > country(String Country) {
        ParkService.ParksImplPort obj =
            new ParkService.ParksImplPort();
        return obj.byCountry(Country);
    }
}

```

ParkLocator Test

```

@isTest
private class ParkLocatorTest {
    @isTest static void testCallout() {
        Test.setMock(WebServiceMock.class, new ParkServiceMock());
        List < String > result = ParkLocator.country('Test');
    }
}

```

Apex Web Services--

AccountManager

```

@RestResource(urlMapping='/Accounts/*/contacts')
global class AccountManager {
    @HttpGet
    global static Account getAccount() {
        RestRequest req = RestContext.request;
        String accId = req.requestURI.substringBetween('Accounts/', '/contacts');
        Account acc = [SELECT Id, Name, (SELECT Id, Name FROM Contacts)
                      FROM Account WHERE Id = :accId];
        return acc;
    }
}

```

```
}
```

AccountManager Test

```
@isTest
private class AccountManagerTest {

    private static testMethod void getAccountTest1() {
        Id recordId = createTestRecord();
        // Set up a test request
        RestRequest request = new RestRequest();
        request.requestUri = 'https://na1.salesforce.com/services/apexrest/Accounts/' +
            recordId +'/contacts';
        request.httpMethod = 'GET';
        RestContext.request = request;
        // Call the method to test
        Account thisAccount = AccountManager.getAccount();
        // Verify results
        System.assert(thisAccount != null);
        System.assertEquals('Test record', thisAccount.Name);

    }

    // Helper method
    static Id createTestRecord() {
        // Create test record
        Account TestAcc = new Account(
            Name='Test record');
        insert TestAcc;
        Contact TestCon= new Contact(
            LastName='Test',
            AccountId = TestAcc.id);
        return TestAcc.Id;
    }
}
```

Apex Testing-- Get Started with Apex Unit Tests

VerifyDate-

```

public class VerifyDate {

    //method to handle potential checks against two dates
    public
    static Date CheckDates(Date date1, Date date2) {

        //if date2 is within the next 30 days of date1, use date2. Otherwise use the end of the month

        if(DateWithin30Days(date1,date2)) {

            return date2;

        } else {

            return SetEndOfMonthDate(date1);

        }
    }

    //method to check if date2 is within the next 30 days of date1

    @TestVisible private static Boolean DateWithin30Days(Date date1, Date date2) {

        //check for date2 being in the past
        if(
            date2 < date1) { return false; }

        //check that date2 is within (>=) 30 days of date1
        Date
        date30Days = date1.addDays(30); //create a date 30 days away from date1

        if( date2 >= date30Days ) { return false; }

        else { return true; }
    }
}

```

```

//method to return the end of the month of a given date

@TestVisible private static Date SetEndOfMonthDate(Date date1) {

Integer totalDays = Date.daysInMonth(date1.year(), date1.month());

Date lastDay = Date.newInstance(date1.year(), date1.month(), totalDays);

return lastDay;
}

}

```

Test VerifyDate

```

@isTest
private class TestVerifyDate{

@isTest static void Test_CheckDates_case1(){
    Date D=VerifyDate.CheckDates(date.parse('01/01/2020'),date.parse('01/05/2020'));
    System.assertEquals(date.parse('01/05/2020'),D);
}

@isTest static void Test_CheckDates_case2(){
    Date D=VerifyDate.CheckDates(date.parse('01/01/2020'),date.parse('05/05/2020'));
    System.assertEquals(date.parse('01/31/2020'),D);
}

@isTest static void Test_DateWithin30Days_case1(){
    Boolean
flag=VerifyDate.DateWithin30Days(date.parse('01/01/2020'),date.parse('12/30/2019'));
    System.assertEquals(false,flag);
}

@isTest static void Test_DateWithin30Days_case2(){
    Boolean flag =
VerifyDate.DateWithin30Days(date.parse('01/01/2020'),date.parse('02/02/2019'));
    System.assertEquals(false,flag);
}

```

```

@isTest static void Test_DateWithin30Days_case3(){

Boolean flag=VerifyDate.DateWithin30Days(date.parse('01/01/2020'),date.parse('01/15/2020'));

System.assertEquals(true,flag);

}

@isTest static void Test_SetEndOfMonthDate(){
    Date returndate=VerifyDate.SetEndOfMonthDate(date.parse('01/01/2020'));
}

}

```

Test Apex Triggers Units

RestrictContactByName-

```

trigger RestrictContactByName on Contact (before insert, before update) {

//check contacts prior to insert or update for invalid data
    For
    (Contact c : Trigger.New) {

        if(c.LastName == 'INVALIDNAME') {
            //invalidname is invalid

            c.AddError('The Last Name "'+c.LastName+'" is not allowed for DML');

        }

    }
}

```

TestRestrictContactByName

```
@isTest
public class TestRestrictContactByName {

    @isTest static void Test_insertupdateContact() {
        Contact cnt=new Contact();
        cnt.LastName='INVALIDNAME';

        Test.startTest();
        Database.SaveResult result = Database.insert(cnt,false);
        Test.stopTest();

        System.assert(!result.isSuccess());
        System.assert(result.getErrors().size()>0);
        System.assertEquals('The Last Name"INVALIDNAME" is not allowed for
DML',result.getErrors()[0].getMessage());
    }
}
```

Create Test Data for Apex Tests Unit

RandomAccountContactFactory-

```
public class RandomAccountContactFactory {

    public static List<Contact> generateRandomContacts (Integer numContacts, String
lastName,Id acclId){
        List<Contact> contacts = new List<Contact>();

        for(integer i = 0; i<numContacts; i++){
            Contact c = new Contact();
            c.FirstName = 'Trail' + i;
            c.LastName = lastName + i;
            c.AccountId = acclId;

            contacts.add(c);
        }
        return contacts;
    }
}
```

```

public static List<Account> generateRandomAccounts (Integer numAccounts){
    List<Account> accounts = new List<Account>();

    for(integer i = 0; i<numAccounts; i++){
        Account a = new Account();
        a.Name = 'Test' + i;
        accounts.add(a);
    }
    return accounts;
}

}

```

APEX SPECIALIST

Maintenance Request Field

Type

[Back to Maintenance Request Fields](#)

[Edit](#) [Set Field-Level Security](#) [View Field Accessibility](#)

Field Information

Field Label	Type	Field Name	Type
Data Type	Picklist		
Help Text			
Description			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			

Picklist Values Used

Active picklist values	5 (1,000 max)
Inactive picklist values	2

Maintenance Request

Validation Rules

No validation rules defined.

Case Type Picklist Values

Action	Values	API Name	Default	Chart Colors	Modified By
Edit Del Deactivate	Mechanical	Mechanical	<input type="checkbox"/>	Assigned dynamically	Korapala Sushma, 6/11/2022, 8:01 AM
Edit Del Deactivate	Electrical	Electrical	<input type="checkbox"/>	Assigned dynamically	Korapala Sushma, 6/11/2022, 8:01 AM
Edit Del Deactivate	Electronic	Electronic	<input type="checkbox"/>	Assigned dynamically	Korapala Sushma, 6/11/2022, 8:01 AM
Edit Del Deactivate	Structural	Structural	<input type="checkbox"/>	Assigned dynamically	Korapala Sushma, 6/11/2022, 8:01 AM
Edit Del Deactivate	Other	Other	<input type="checkbox"/>	Assigned dynamically	Korapala Sushma, 6/11/2022, 8:01 AM

Inactive Values

Action	Values	API Name	Modified By
Del Activate	Routine Maintenance	Routine Maintenance	Korapala Sushma, 6/11/2022, 8:44 AM
Del Activate	Inspection'	Inspection'	Korapala Sushma, 6/11/2022, 8:47 AM

Equipment

Custom Field Definition Detail

Field Label	Field Name	Object Name
Equipment	Equipment	Case
	Equipment	Data Type
	Equipment_c	Lookup
Description		
Help Text		
Data Owner		
Field Usage		
Data Sensitivity Level		
Compliance Categorization		
Created By	Korapala Sushma, 6/11/2022, 8:35 AM	Modified By
		Korapala Sushma, 6/11/2022, 8:35 AM

MaintenanceRequest

```
trigger MaintenanceRequest on Case (before update, after update) {
    if(Trigger.isUpdate && Trigger.isAfter){
        MaintenanceRequestHelper.updateWorkOrders(Trigger.New, Trigger.OldMap);
    }
}
```

MaintenanceRequestHelper

```

public with sharing class MaintenanceRequestHelper {
    public static void updateWorkOrders(List<Case> updWorkOrders, Map<Id, Case>
nonUpdCaseMap) {
        Set<Id> validIds = new Set<Id>();
        For (Case c : updWorkOrders){
            if (nonUpdCaseMap.get(c.Id).Status != 'Closed' && c.Status == 'Closed'){
                if (c.Type == 'Repair' || c.Type == 'Routine Maintenance'){
                    validIds.add(c.Id);
                }
            }
        }

        //When an existing maintenance request of type Repair or Routine Maintenance is closed,
        //create a new maintenance request for a future routine checkup.
        if (!validIds.isEmpty()){
            Map<Id, Case> closedCases = new Map<Id, Case>([SELECT Id, Vehicle__c, Equipment__c,
Equipment__r.Maintenance_Cycle__c,
(SELECT Id, Equipment__c, Quantity__c FROM
Equipment_Maintenance_Items__r)
FROM Case WHERE Id IN :validIds]);
            Map<Id, Decimal> maintenanceCycles = new Map<ID, Decimal>();

            //calculate the maintenance request due dates by using the maintenance cycle defined
            on the related equipment records.
            AggregateResult[] results = [SELECT Maintenance_Request__c,
                MIN(Equipment__r.Maintenance_Cycle__c)cycle
                FROM Equipment_Maintenance_Item__c
                WHERE Maintenance_Request__c IN :ValidIds GROUP BY
Maintenance_Request__c];

            for (AggregateResult ar : results){
                maintenanceCycles.put((Id) ar.get('Maintenance_Request__c'), (Decimal)
ar.get('cycle'));
            }

            List<Case> newCases = new List<Case>();
            for(Case cc : closedCases.values()){
                Case nc = new Case (
                    ParentId = cc.Id,
                    Status = 'New',

```

```

        Subject = 'Routine Maintenance',
        Type = 'Routine Maintenance',
        Vehicle__c = cc.Vehicle__c,
        Equipment__c = cc.Equipment__c,
        Origin = 'Web',
        Date_Reported__c = Date.Today()
    );

    //If multiple pieces of equipment are used in the maintenance request,
    //define the due date by applying the shortest maintenance cycle to today's date.
    //If (maintenanceCycles.containsKey(cc.Id)){
        nc.Date_Due__c = Date.today().addDays((Integer) maintenanceCycles.get(cc.Id));
    //}
    // nc.Date_Due__c = Date.today().addDays((Integer)
cc.Equipment__r.maintenance_Cycle__c);
}

newCases.add(nc);
}

```

WarehouseCalloutService

```

public with sharing class WarehouseCalloutService implements Queueable {
    private static final String WAREHOUSE_URL = 'https://th-superbadge-
apex.herokuapp.com/equipment';

    //Write a class that makes a REST callout to an external warehouse system to get a list of
    equipment that needs to be updated.
    //The callout's JSON response returns the equipment records that you upsert in Salesforce.

    @future(callout=true)
    public static void runWarehouseEquipmentSync(){
        System.debug('go into runWarehouseEquipmentSync');
        Http http = new Http();
        HttpRequest request = new HttpRequest();

```

```

request.setEndpoint(WAREHOUSE_URL);
request.setMethod('GET');
HttpResponse response = http.send(request);

List<Product2> product2List = new List<Product2>();
System.debug(response.getStatusCode());
if (response.getStatusCode() == 200){
    List<Object> jsonResponse =
(List<Object>)JSON.deserializeUntyped(response.getBody());
System.debug(response.getBody())
//class maps the following fields:
    //warehouse SKU will be external ID for identifying which equipment records to update
within Salesforce
    for (Object jR : jsonResponse){
        Map<String, Object> mapJson = (Map<String, Object>)jR;
        Product2 product2 = new Product2();
        //replacement part (always true),
        product2.Replacement_Part__c = (Boolean) mapJson.get('replacement');
        //cost
        product2.Cost__c = (Integer) mapJson.get('cost');
        //current inventory
        product2.Current_Inventory__c = (Double) mapJson.get('quantity');
        //lifespan
        product2.Lifespan_Months__c = (Integer) mapJson.get('lifespan');
        //maintenance cycle
        product2.Maintenance_Cycle__c = (Integer) mapJson.get('maintenanceperiod');
        //warehouse SKU
        product2.Warehouse_SKU__c = (String) mapJson.get('sku');

        product2.Name = (String) mapJson.get('name');
        product2.ProductCode = (String) mapJson.get('_id');
        product2List.add(product2);
    }
}

if (product2List.size() > 0){
    upsert product2List;
    System.debug('Your equipment was synced with the warehouse one');
}
}
}

```

```
public static void execute (QueueableContext context){  
    System.debug('start runWarehouseEquipmentSync');  
    runWarehouseEquipmentSync();  
    System.debug('end runWarehouseEquipmentSync');  
}  
}
```

WarehouseCalloutServiceTest

```
@IsTest  
private class WarehouseCalloutServiceTest {  
    // implement your mock callout test here  
  
@isTest  
static void testWarehouseCallout() {  
    test.startTest();  
    test.setMock(HttpCalloutMock.class, new WarehouseCalloutServiceMock());  
    WarehouseCalloutService.execute(null);  
    test.stopTest();  
  
    List<Product2> product2List = new List<Product2>();  
    product2List = [SELECT ProductCode FROM Product2];  
  
    System.assertEquals(3, product2List.size());  
    System.assertEquals('55d66226726b611100aaf741', product2List.get(0).ProductCode);  
    System.assertEquals('55d66226726b611100aaf742', product2List.get(1).ProductCode);  
    System.assertEquals('55d66226726b611100aaf743', product2List.get(2).ProductCode);  
}  
}
```

WarehouseSyncSchedule

```
global class WarehouseSyncSchedule implements Schedulable {  
    global void execute(SchedulableContext ctx) {
```

```

        WarehouseCalloutService.runWarehouseEquipmentSync();
    }
}

WarehouseSyncScheduleTest

@isTest
public class WarehouseSyncScheduleTest {

    @isTest static void WarehousescheduleTest(){
        String scheduleTime = '00 00 01 * * ?';
        Test.startTest();
        Test.setMock(HttpCalloutMock.class, new WarehouseCalloutServiceMock());
        String jobID=System.schedule('Warehouse Time To Schedule to Test', scheduleTime, new
WarehouseSyncSchedule());
        Test.stopTest();
        //Contains schedule information for a scheduled job. CronTrigger is similar to a cron job on
UNIX systems.
        // This object is available in API version 17.0 and later.
        CronTrigger a=[SELECT Id FROM CronTrigger where NextFireTime > today];
        System.assertEquals(jobID, a.Id,'Schedule ');
    }

}

```

MaintenanceRequestHelperTest

```

@isTest
public with sharing class MaintenanceRequestHelperTest {

    // createVehicle
    private static Vehicle__c createVehicle(){
        Vehicle__c vehicle = new Vehicle__C(name = 'Testing Vehicle');
        return vehicle;
    }

    // createEquipment
    private static Product2 createEquipment(){

```

```

product2 equipment = new product2(name = 'Testing equipment',
                                 lifespan_months__c = 10,
                                 maintenance_cycle__c = 10,
                                 replacement_part__c = true);
return equipment;
}

// createMaintenanceRequest
private static Case createMaintenanceRequest(id vehicleId, id equipmentId){
    case cse = new case(Type='Repair',
                         Status='New',
                         Origin='Web',
                         Subject='Testing subject',
                         Equipment__c=equipmentId,
                         Vehicle__c=vehicleId);
    return cse;
}

// createEquipmentMaintenanceItem
private static Equipment_Maintenance_Item__c createEquipmentMaintenanceItem(id
equipmentId,id requestId){
    Equipment_Maintenance_Item__c equipmentMaintenanceItem = new
Equipment_Maintenance_Item__c(
    Equipment__c = equipmentId,
    Maintenance_Request__c = requestId);
    return equipmentMaintenanceItem;
}

@isTest
private static void testPositive(){
    Vehicle__c vehicle = createVehicle();
    insert vehicle;
    id vehicleId = vehicle.Id;

    Product2 equipment = createEquipment();
    insert equipment;
    id equipmentId = equipment.Id;

    case createdCase = createMaintenanceRequest(vehicleId,equipmentId);
    insert createdCase;
}

```

```
Equipment_Maintenance_Item__c equipmentMaintenanceItem =
createEquipmentMaintenanceItem(equipmentId, createdCase.id);
insert equipmentMaintenanceItem;

test.startTest();
createdCase.status = 'Closed';
update createdCase;
test.stopTest();
```

```
Case newCase = [Select id,
                subject,
                type,
                Equipment__c,
                Date_Reported__c,
                Vehicle__c,
                Date_Due__c
                from case
                where status ='New'];
```

```
Equipment_Maintenance_Item__c workPart = [select id
                                              from Equipment_Maintenance_Item__c
                                              where Maintenance_Request__c =:newCase.Id];
list<case> allCase = [select id from case];
system.assert(allCase.size() == 2);

system.assert(newCase != null);
system.assert(newCase.Subject != null);
system.assertEquals(newCase.Type, 'Routine Maintenance');
SYSTEM.assertEquals(newCase.Equipment__c, equipmentId);
SYSTEM.assertEquals(newCase.Vehicle__c, vehicleId);
SYSTEM.assertEquals(newCase.Date_Reported__c, system.today());
}
```

```
@isTest
private static void testNegative(){
    Vehicle__C vehicle = createVehicle();
    insert vehicle;
    id vehicleId = vehicle.Id;
```

```

product2 equipment = createEquipment();
insert equipment;
id equipmentId = equipment.Id;

case createdCase = createMaintenanceRequest(vehicleId,equipmentId);
insert createdCase;

Equipment_Maintenance_Item__c workP = createEquipmentMaintenanceItem(equipmentId,
createdCase.Id);
insert workP;

test.startTest();
createdCase.Status = 'Working';
update createdCase;
test.stopTest();

list<case> allCase = [select id from case];

Equipment_Maintenance_Item__c equipmentMaintenanceItem = [select id
from Equipment_Maintenance_Item__c
where Maintenance_Request__c = :createdCase.Id];

system.assert(equipmentMaintenanceItem != null);
system.assert(allCase.size() == 1);
}

@isTest
private static void testBulk(){
list<Vehicle__C> vehicleList = new list<Vehicle__C>();
list<Product2> equipmentList = new list<Product2>();
list<Equipment_Maintenance_Item__c> equipmentMaintenanceItemList = new
list<Equipment_Maintenance_Item__c>();
list<case> caseList = new list<case>();
list<id> oldCaselds = new list<id>();

for(integer i = 0; i < 300; i++){
    vehicleList.add(createVehicle());
    equipmentList.add(createEquipment());
}
}

```

```

insert vehicleList;
insert equipmentList;

for(integer i = 0; i < 300; i++){
    caseList.add(createMaintenanceRequest(vehicleList.get(i).id, equipmentList.get(i).id));
}
insert caseList;

for(integer i = 0; i < 300; i++){

equipmentMaintenanceItemList.add(createEquipmentMaintenanceItem(equipmentList.get(i).id,
caseList.get(i).id));
}
insert equipmentMaintenanceItemList;

test.startTest();
for(case cs : caseList){
    cs.Status = 'Closed';
    oldCaselds.add(cs.Id);
}
update caseList;
test.stopTest();

list<case> newCase = [select id
                        from case
                        where status ='New'];

list<Equipment_Maintenance_Item__c> workParts = [select id
                                                from Equipment_Maintenance_Item__c
                                                where Maintenance_Request__c in: oldCaselds];

system.assert(newCase.size() == 300);

list<case> allCase = [select id from case];
system.assert(allCase.size() == 600);
}

}

```

WarehouseCalloutServiceMock

```
@isTest
global class WarehouseCalloutServiceMock implements HttpCalloutMock {
    // implement http mock callout
    global static HttpResponse respond(HttpRequest request) {

        HttpResponse response = new HttpResponse();
        response.setHeader('Content-Type', 'application/json');

        response.setBody('[{"_id":"55d66226726b611100aaf741","replacement":false,"quantity":5,"name":
"Generator 1000
kW","maintenanceperiod":365,"lifespan":120,"cost":5000,"sku":"100003"}, {"_id":"55d66226726b611
100aaf742","replacement":true,"quantity":183,"name":"Cooling
Fan","maintenanceperiod":0,"lifespan":0,"cost":300,"sku":"100004"}, {"_id":"55d66226726b611100a
af743","replacement":true,"quantity":143,"name":"Fuse
20A","maintenanceperiod":0,"lifespan":0,"cost":22,"sku":"100005"}]');
        response.setStatusCode(200);

        return response;
    }
}
```

WarehouseCalloutServiceTest

```
@IsTest
private class WarehouseCalloutServiceTest {
    // implement your mock callout test here

@isTest
    static void testWarehouseCallout() {
        test.startTest();
        test.setMock(HttpCalloutMock.class, new WarehouseCalloutServiceMock());
        WarehouseCalloutService.execute(null);
        test.stopTest();

        List<Product2> product2List = new List<Product2>();
        product2List = [SELECT ProductCode FROM Product2];
```

```

        System.assertEquals(3, product2List.size());
        System.assertEquals('55d66226726b611100AAF741', product2List.get(0).ProductCode);
        System.assertEquals('55d66226726b611100AAF742', product2List.get(1).ProductCode);
        System.assertEquals('55d66226726b611100AAF743', product2List.get(2).ProductCode);
    }
}

```

WarehouseSyncSchedule

```

global class WarehouseSyncSchedule implements Schedulable {
    global void execute(SchedulableContext ctx) {

        WarehouseCalloutService.runWarehouseEquipmentSync();
    }
}

```

WarehouseSyncScheduleTest

```

@isTest
public class WarehouseSyncScheduleTest {

    @isTest static void WarehousescheduleTest(){
        String scheduleTime = '00 00 01 * * ?';
        Test.startTest();
        Test.setMock(HttpCalloutMock.class, new WarehouseCalloutServiceMock());
        String jobID=System.schedule('Warehouse Time To Schedule to Test', scheduleTime, new
WarehouseSyncSchedule());
        Test.stopTest();
        //Contains schedule information for a scheduled job. CronTrigger is similar to a cron job on
UNIX systems.
        // This object is available in API version 17.0 and later.
        CronTrigger a=[SELECT Id FROM CronTrigger WHERE NextFireTime > today];
        System.assertEquals(jobID, a.Id,'Schedule ');
    }
}

```

}

Process Automation Specialist

Validations and Formulas

Use Formula Field

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes 'Setup', 'Home', 'Object Manager', and various global icons. The main area displays the 'Contract' object details. On the left, a sidebar lists layout options like 'Page Layouts', 'Lightning Record Pages', and 'Buttons, Links, and Actions'. The central panel shows the 'Fields & Relationships' section for a custom field named 'Days Remaining'. The 'Custom Field Definition Detail' page includes fields for 'Field Label' (Days Remaining), 'Field Name' (Days_Remaining), 'API Name' (Days_Remaining__c), 'Object Name' (Contract), 'Data Type' (Formula), and 'Decimal Places' (2). The formula used is 'EndDate - TODAY()'. Navigation links include 'Edit', 'Set Field-Level Security', 'View Field Accessibility', and 'Where is this used?'. A help icon and a 'Help for this Page' link are also present.

Implement Rollup Summary Fields

The screenshot shows the Salesforce Setup interface for creating a custom field. The top navigation bar includes 'Setup', 'Home', 'Object Manager', a search bar, and various system icons. The main title is 'SETUP > OBJECT MANAGER Account'. On the left, a sidebar lists options like 'Details', 'Fields & Relationships' (which is selected), and various layout-related items. The right pane displays the 'Custom Field Definition Detail' for 'Potential Value'. Key details shown include:

Field Label	Potential Value
Field Name	Potential_Value
API Name	Potential_Value_c
Description	(empty)
Help Text	(empty)
Data Owner	(empty)
Field Usage	(empty)
Data Sensitivity Level	(empty)
Compliance Categorization	(empty)
Created By	Korapala Sushma, 6/4/2022, 10:18 AM
Modified By	Korapala Sushma, 6/4/2022, 10:18 AM

Below this, the 'Roll-Up Summary Options' section is visible, showing 'Data Type: Roll-Up Summary', 'Summarized Object: Opportunity', 'Field to Aggregate: Opportunity_Expected_Revenue', and 'Summary Type: SUM'.

Create a Validation Rule

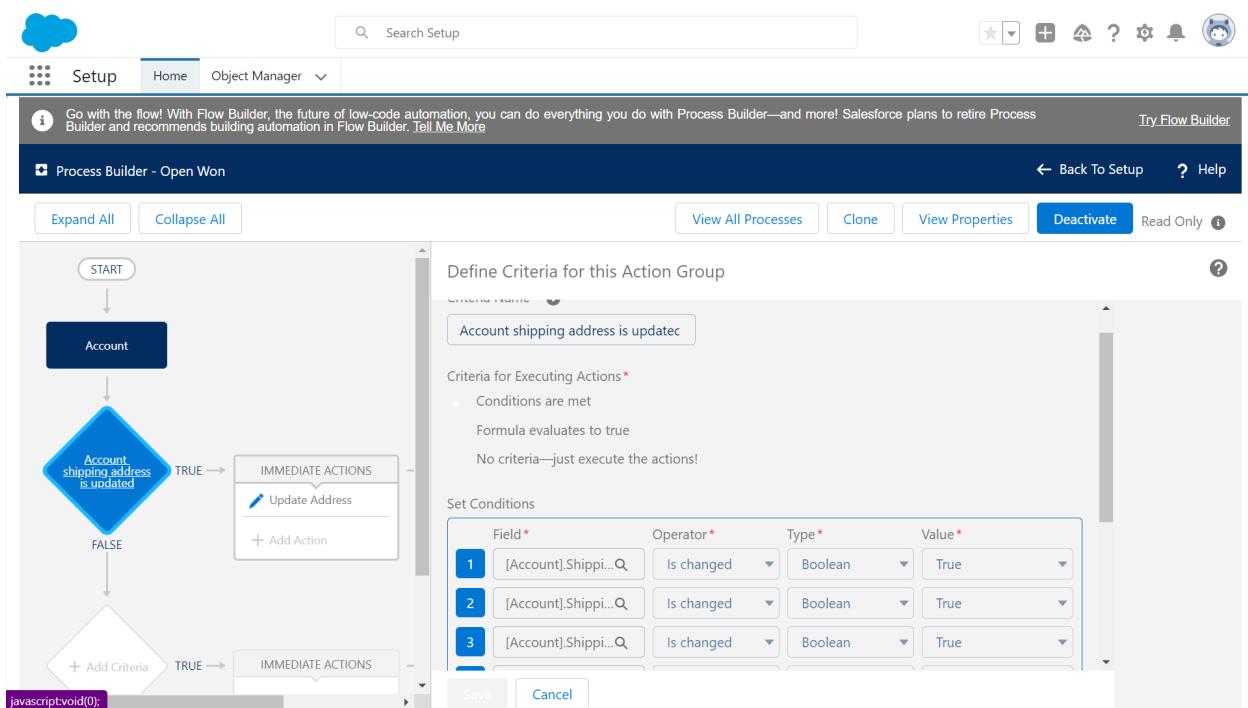
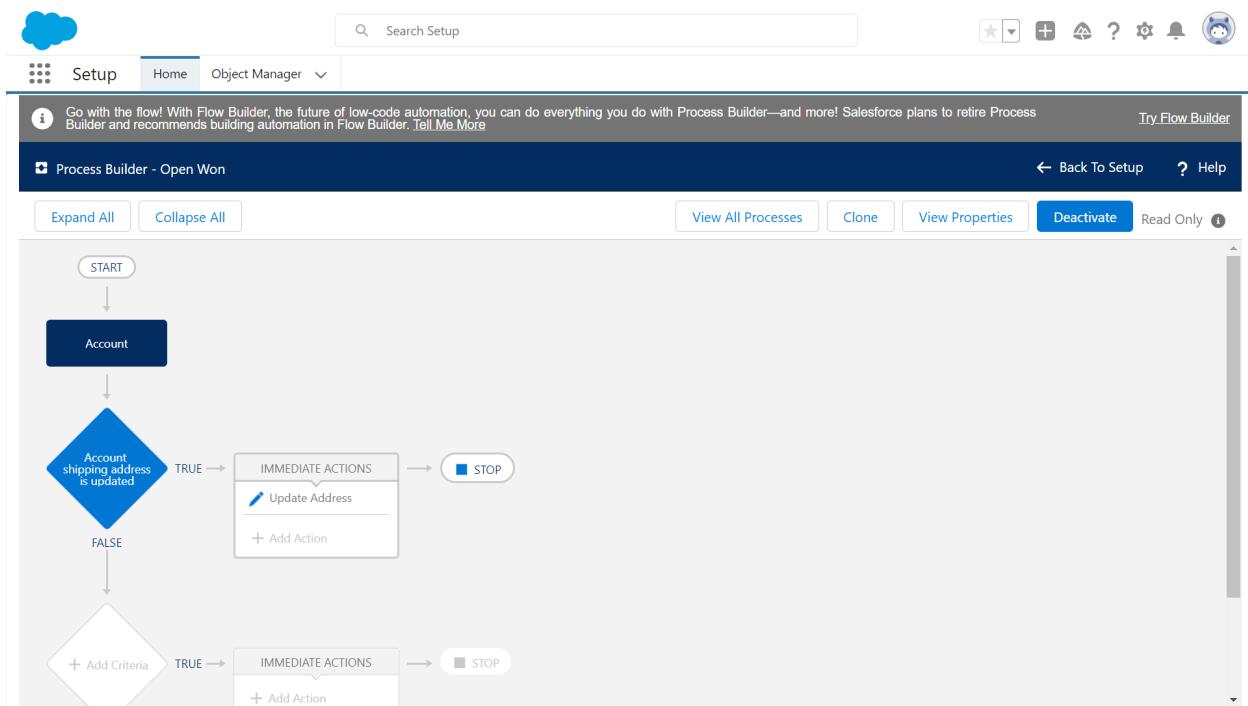
The screenshot shows the Salesforce Setup interface for creating a validation rule. The top navigation bar includes 'Setup', 'Home', 'Object Manager', a search bar, and various system icons. The main title is 'SETUP > OBJECT MANAGER Contact'. On the left, a sidebar lists options like 'Details', 'Fields & Relationships' (which is selected), and various layout-related items. The right pane displays the 'Validation Rule Detail' for 'Contact must be in Account ZIP Code'. Key details shown include:

Rule Name	Contact_must_be_in_Account_ZIP_Code
Error Condition Formula	AND(NOT(ISBLANK(AccountId)), MailingPostalCode <> Account.ShippingPostalCode)
Error Message	Contact must be in Account ZIP Code
Description	(empty)
Created By	Korapala Sushma, 6/4/2022, 10:28 AM
Modified By	Korapala Sushma, 6/4/2022, 10:28 AM

The 'Active' checkbox is checked. The 'Error Location' is set to 'Top of Page'.

Salesforce Flow

Automate Simple Business Processes with process builder Unit



Setup Home Object Manager

Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with Process Builder—and more! Salesforce plans to retire Process Builder and recommends building automation in Flow Builder. [Tell Me More](#)

[Try Flow Builder](#)

Process Builder - Open Won

← Back To Setup ? Help

Expand All Collapse All

View All Processes Clone View Properties Deactivate Read Only

The screenshot shows a process flow starting with a 'START' state, followed by an 'Account' object state. An 'Account shipping address is updated' decision diamond follows. If 'TRUE', it leads to an 'IMMEDIATE ACTIONS' section containing an 'Update Address' action. If 'FALSE', it leads to another decision diamond '+ Add Criteria'. This second decision diamond has a 'TRUE' path leading to another 'IMMEDIATE ACTIONS' section with an 'Update Address' action, and a 'STOP' state. A 'Save' button is at the bottom.

Define Criteria for this Action Group

Field *	Operator *	Type *	Value *
[Account].ShippingCity	Is changed	Boolean	True
[Account].ShippingCity	Is changed	Boolean	True
[Account].ShippingCity	Is changed	Boolean	True
[Account].ShippingCity	Is changed	Boolean	True
[Account].ShippingCity	Is changed	Boolean	True

Conditions *

- All of the conditions are met (AND)
- Any of the conditions are met (OR)
- Customize the logic

Save Cancel

Setup Home Object Manager

Go with the flow! With Flow Builder, the future of low-code automation, you can do everything you do with Process Builder—and more! Salesforce plans to retire Process Builder and recommends building automation in Flow Builder. [Tell Me More](#)

[Try Flow Builder](#)

Process Builder - Open Won

← Back To Setup ? Help

Expand All Collapse All

View All Processes Clone View Properties Deactivate Read Only

The screenshot shows a process flow starting with a 'START' state, followed by an 'Account' object state. An 'Account shipping address is updated' decision diamond follows. If 'TRUE', it leads to an 'IMMEDIATE ACTIONS' section containing an 'Update Address' action, which is highlighted with a blue border. If 'FALSE', it leads to another decision diamond '+ Add Criteria'. This second decision diamond has a 'TRUE' path leading to another 'IMMEDIATE ACTIONS' section with an 'Update Address' action, and a 'STOP' state. A 'Save' button is at the bottom.

Update Records

Action Name *

Record *

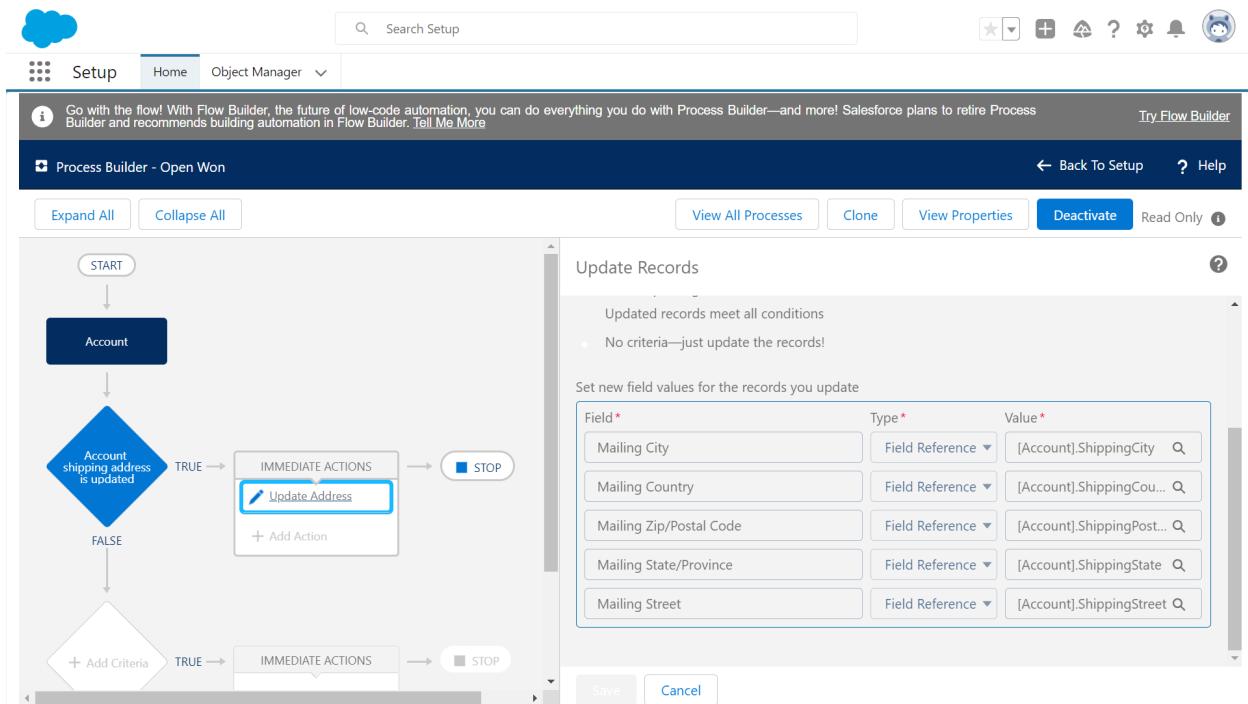
Criteria for Updating Records *

- Updated records meet all conditions
- No criteria—just update the records!

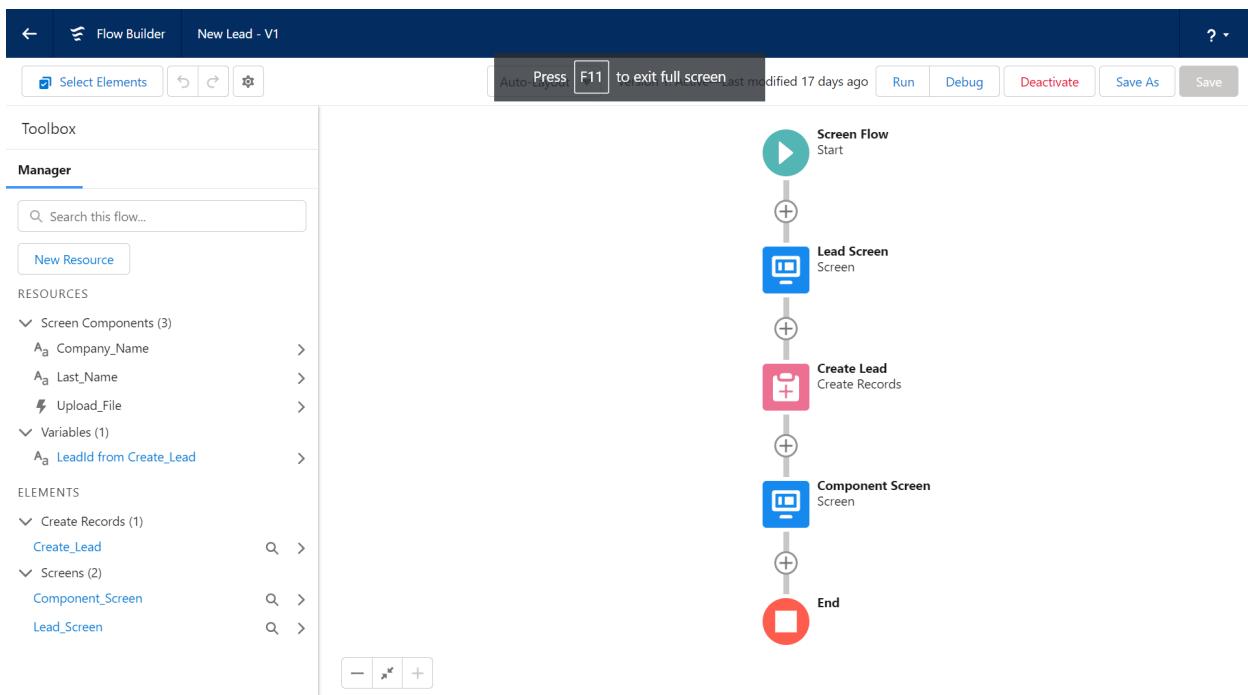
Set new field values for the records you update

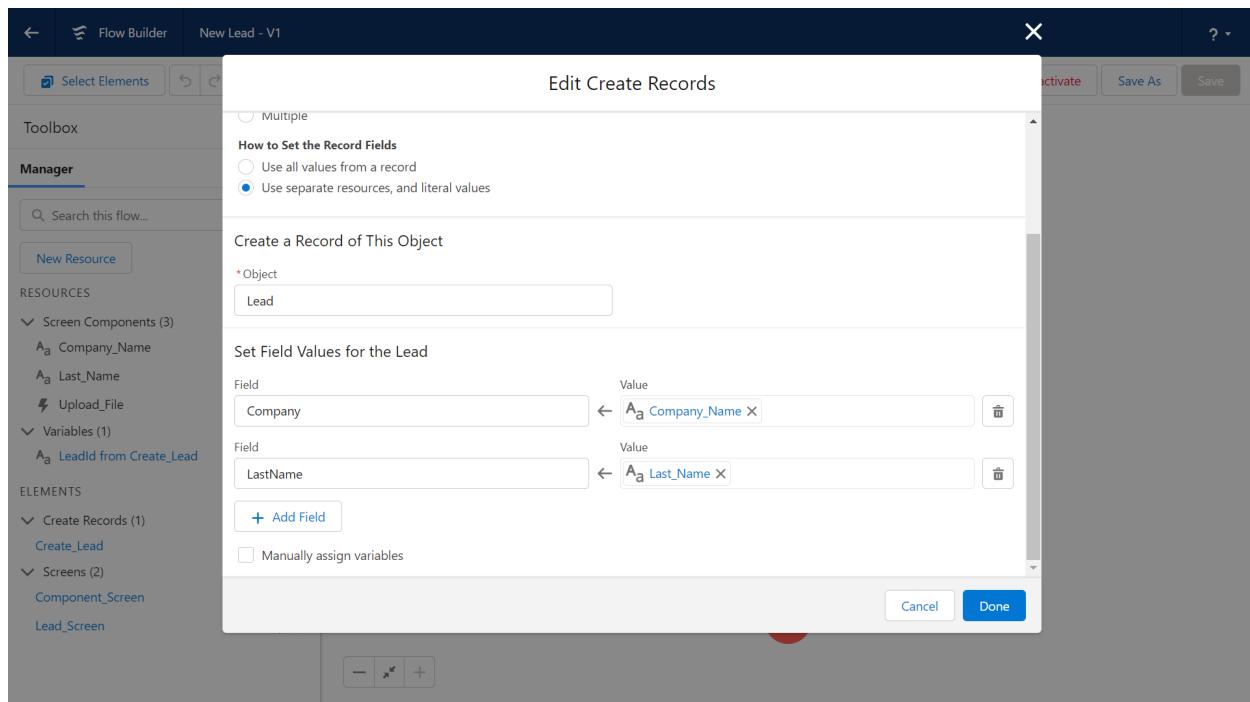
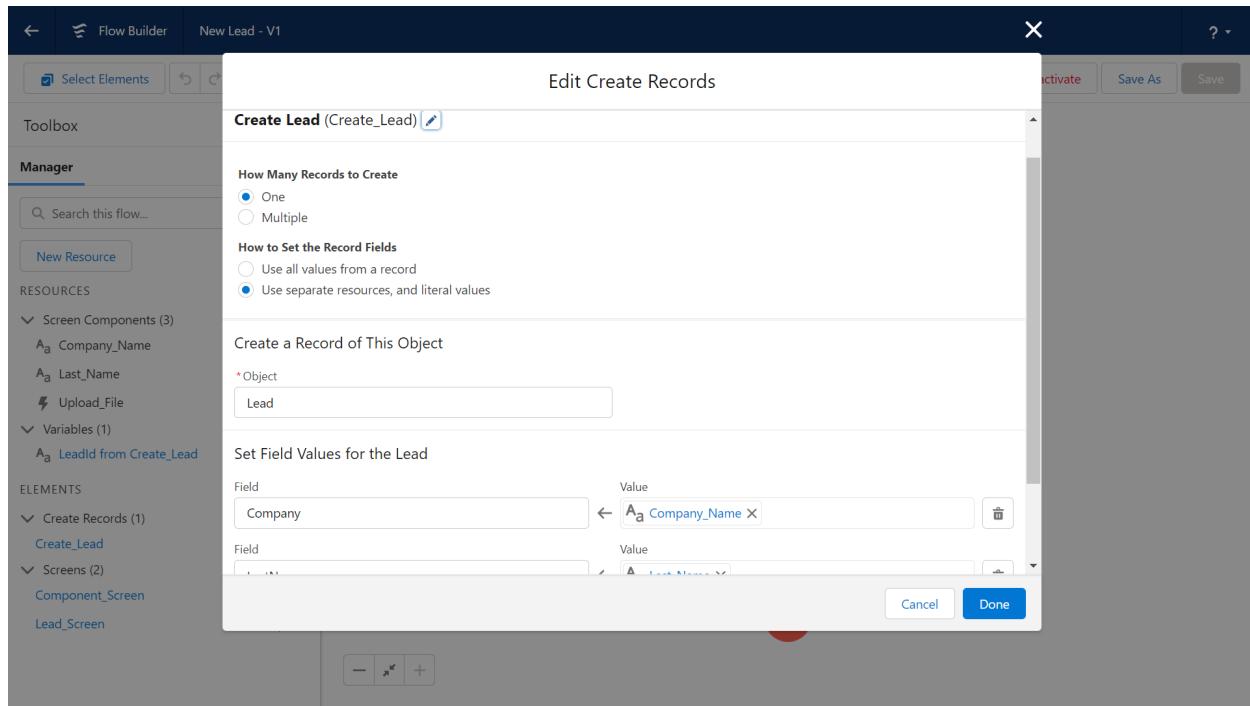
Field *	Type *	Value *
Mailing City	Field Reference	[Account].ShippingCity

Save Cancel



Guide Users Through Your Business Processes with Flow Builder





Customize How Records Get Approved with Approvals

The screenshot shows the Salesforce Setup interface. The left sidebar navigation includes 'Setup', 'Home', 'Object Manager', and sections for 'Apps', 'Workflow Services', 'Process Automation', 'Workflow Actions', and 'Environments'. The main content area is titled 'Approval Processes' and shows a process named 'Approve New Account' for 'Account: Approve New Account'. The 'Process Definition Detail' section includes fields for Process Name ('Approve New Account'), Unique Name ('Approve_New_Account'), Description ('(Account: Type EQUALS Prospect) AND (Account: Employees GREATER THAN 500)'), Entry Criteria ('Administrator ONLY'), Record Editability ('Allow Submitters to Recall Approval Requests'), Approval Assignment Email Template ('Marketing_Product Inquiry Response'), Initial Submitters ('Account Owner'), and Created By ('Korapala Sushma'). The 'Initial Submission Actions' section contains a single entry: 'Record Lock' (Description: 'Lock the record from being edited'). The 'Approval Steps' section is currently empty.

This screenshot shows the same Approval Processes page as the previous one, but with more detailed configurations. The 'Approval Steps' section now lists a single step named 'Step 1' assigned to 'User Korapala Sushma' with a 'Final Rejection' reject behavior. The 'Final Approval Actions' section contains two entries: 'Record Lock' (Description: 'Lock the record from being edited') and 'Field Update' (Description: 'Account Type To Customer'). The 'Final Rejection Actions' section contains two entries: 'Record Lock' (Description: 'Unlock the record for editing') and 'Field Update' (Description: 'Account Type To Prospect'). The 'Recall Actions' section contains a single entry: 'Record Lock' (Description: 'Unlock the record for editing'). The bottom of the page includes a 'Back To Top' link and a note about showing more records per related list.

Leads & Opportunities for Lightning Experience

Create and Convert Leads as Potential Customers

The screenshot shows the Salesforce Lightning Experience interface for a Lead record. The top navigation bar includes a blue cloud icon, a search bar with placeholder "Search Leads and more...", and various global buttons like "Follow", "Convert", "Edit", and "New Case". The main header displays the Lead's name, "Mr. Chantal Smith". Below the header, a status bar shows the lead's current status: "Working - Contacted". To the right of the status bar are buttons for "Mark Status as Complete" and other actions. The main content area is divided into sections: "Activity" (with tabs for Log a Call, Details, Chatter, and News), "Upcoming & Overdue" tasks (one task listed: "Call Ms. Smith"), and a "Related" section which states "We found no potential duplicates of this Lead". A note indicates that no duplicate rules are activated. The bottom of the page shows a message: "No past activity. Past meetings and tasks marked as done show up here."

Work Your Opportunities

Cloudy

How We Roll Maint... Products Cases Equipment Maintenance Items Vehicles * Get Cloudy - 24 Holiday S... * More

Opportunity
Get Cloudy - 24 Holiday Sneakers

Account Name: Get Cloudy Close Date: 6/30/2022 Amount: Opportunity Owner: Korapala Sushma

+ Follow Edit New Case New Note

Value Propo... Id. Decision ... Perception ... Proposal/Pri... Negotiation... Closed ✓ Mark Stage as Complete

Activity Details Chatter

Opportunity Owner  Korapala Sushma	Amount
Private	Expected Revenue
<input type="checkbox"/>	Close Date 6/30/2022
Opportunity Name Get Cloudy - 24 Holiday Sneakers	Next Step
Account Name Get Cloudy	Stage
Type New Customer	Value Proposition
Lead Source Web	Probability (%) 50%
Discount Percentage	Primary Campaign Source

Related

- Products (0)
- Notes & Attachments (0)
Upload Files
Or drop files
- Contact Roles (1)
 Alan Johnson
Role: Decision Maker
Title:
View All

Cloudy

How We Roll Maint... Products Cases Equipment Maintenance Items Press F11 to exit full screen * Get Cloudy - 24 Holiday S... * More

Opportunity Owner: Korapala Sushma Amount: Expected Revenue

Private

Opportunity Name: Get Cloudy - 24 Holiday Sneakers Close Date: 6/30/2022

Account Name: Get Cloudy Next Step

Type: New Customer Stage: Value Proposition

Lead Source: Web Probability (%): 50%

Discount Percentage: Primary Campaign Source

Approval Status:

Order Number: Main Competitor(s):

Current Generator(s): Delivery/Installation Status:

Tracking Number:

Created By: Korapala Sushma, 6/6/2022, 9:33 AM Last Modified By: Korapala Sushma, 6/6/2022, 9:33 AM

Products (0)

Notes & Attachments (0)
Upload Files
Or drop files

Contact Roles (1)
 Alan Johnson
Role: Decision Maker
Title:
View All

Partners (0)

Stage History (1)

Process Automation Specialist

Automate Leads

The screenshot shows the 'Lead Validation Rule' setup page in Salesforce. The left sidebar lists various configuration options like Details, Fields & Relationships, Page Layouts, etc. The main content area displays the validation rule details:

Validation Rule Detail	
Rule Name	Anything
Error Condition Formula	OR(AND(LEN(State) > 2, NOT(CONTAINS("AL AK AZ AR CA CO CT DE DC FL GA HI ID IL IN IA KS KY LA ME MD MA MI MN MS MO MT NE NV NH NJ NM NY NC ND OH OK OR PA RI SC SD State)), NOT(OR(Country = "US", Country = "United States", ISBLANK(Country))))
Error Message	We're a US-based company, and for now we can only do business in the United States. I can't do anything with international leads.
Description	
Created By	Korapala Sushma, 6/9/2022, 6:23 AM
Modified By	Korapala Sushma, 6/9/2022, 7:02 AM

The screenshot shows the 'Assembly System Sales' queue setup page in Salesforce. The left sidebar shows navigation links like Setup Home, Service Setup Assistant, etc. The main content area displays the queue details:

Queue		Assembly System Sales	
Label	Assembly System Sales	Queue Name	Assembly_System_Sales
Queue Email		Send Email to Members	<input type="checkbox"/>
Supported Objects	Lead	Created By	Korapala Sushma, 6/9/2022, 7:05 AM
Modified By	Korapala Sushma, 6/9/2022, 7:05 AM	View All Users	
No members.			

Setup Home

Service Setup Assistant

Multi-Factor Authentication Assistant

Release Updates

Lightning Experience Transition Assistant

New Salesforce Mobile App QuickStart

Lightning Usage

Optimizer

ADMINISTRATION

Users

- Permission Set Groups
- Permission Sets
- Profiles
- Public Groups

Queues

Roles

User Management Settings

SETUP

Queues

Queue Rainbow Sales

Help for this Page ?

Label	Rainbow Sales	Queue Name	Rainbow_Sales
Queue Email		Send Email to Members	<input type="checkbox"/>
Supported Objects	Lead	Created By	Korapala Sushma, 6/9/2022, 7:05 AM
		Modified By	Korapala Sushma, 6/9/2022, 7:05 AM

No members.

[View All Users](#)

Setup Home

Object Manager

Q lead

Feature Settings

Marketing

Lead Assignment Rules

Lead Auto-Response Rules

Lead Processes

Lead Settings

LinkedIn Lead Gen

Lead Gen Fields

LinkedIn Accounts

Web-to-Lead

Sales

Social Accounts and Contacts

Settings

SETUP

Lead Assignment Rules

Lead Assignment Rule Anything

Help for this Page ?

Add rule entries that specify the criteria used to route leads. You can reorder rule entries on this page after you create them.

Rule Detail

Rule Name	Anything	Active	<input checked="" type="checkbox"/>
Created By	Korapala Sushma, 6/9/2022, 7:06 AM	Modified By	Korapala Sushma, 6/9/2022, 7:08 AM

Rule Entries

Action	Order	Criteria	Assign To	Email
Edit Del	1	Lead: Lead Source EQUALS Web	Rainbow Sales	<input type="checkbox"/>
Edit Del	2	Lead: Lead Source NOT EQUAL TO Web	Assembly System Sales	<input type="checkbox"/>

Automate Accounts

The screenshot shows the Salesforce Object Manager interface for creating a custom field. The top navigation bar includes a cloud icon, 'Setup', 'Home', and 'Object Manager'. The main title is 'Account Custom Field Number of deals'. The left sidebar under 'Fields & Relationships' lists various layout types. The right panel displays the 'Custom Field Definition Detail' for 'Number of deals', which is a Roll-Up Summary field for the 'Opportunity' object. It has a field label of 'Number of deals', field name 'Number_of_deals', and API name 'Number_of_deals__c'. The 'Object Name' is 'Account'. The 'Data Type' is 'Roll-Up Summary' and the 'Summary Type' is 'COUNT'. The 'Created By' is Korapala Sushma, and the 'Modified By' is also Korapala Sushma.

This screenshot shows the continuation of the custom field creation process. The title is now 'Account Custom Field Number of won deals'. The right panel shows the same field definition details as the previous screenshot, but with a different summary type. The 'Summary Type' is now set to 'COUNT' instead of 'SUM'. The 'Filter Criteria' is 'Stage EQUALS Closed Won'. The rest of the field information remains the same, including the field label, name, API name, object name, and data type.

Setup Home Object Manager Object Manager > Account

Search Setup Press F11 to exit full screen

Last won deal date

Help for this Page

Custom Field Definition Detail

Field Information

Field Label	Last won deal date	Object Name	Account
Field Name	Last_won_deal_date		
API Name	Last_won_deal_date_c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Korapala Sushma, 6/9/2022, 9:00 AM	Modified By	Korapala Sushma, 6/9/2022, 9:00 AM

Roll-Up Summary Options

Data Type	Roll-Up Summary	Summary Type	MAX
Summarized Object	Opportunity		
Field to Aggregate	Opportunity.Close Date		
Filter Criteria	Stage EQUALS Closed Won		

Details Fields & Relationships Page Layouts Lightning Record Pages Buttons, Links, and Actions Compact Layouts Field Sets Object Limits Record Types Related Lookup Filters Search Layouts List View Button Layout Hierarchy Columns

Setup Home Object Manager Object Manager > Account

Search Setup

Amount of won deals

Help for this Page

Custom Field Definition Detail

Field Information

Field Label	Amount of won deals	Object Name	Account
Field Name	Amount_of_won_deals		
API Name	Amount_of_won_deals_c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Korapala Sushma, 6/9/2022, 8:57 AM	Modified By	Korapala Sushma, 6/9/2022, 8:57 AM

Roll-Up Summary Options

Data Type	Roll-Up Summary	Summary Type	SUM
Summarized Object	Opportunity		
Field to Aggregate	Opportunity.Amount		
Filter Criteria	Stage EQUALS Closed Won		

Details Fields & Relationships Page Layouts Lightning Record Pages Buttons, Links, and Actions Compact Layouts Field Sets Object Limits Record Types Related Lookup Filters Search Layouts List View Button Layout Hierarchy Columns

Setup > Object Manager

Account

Details

Fields & Relationships

- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Search Layouts
- List View Button Layout
- Hierarchy Columns

Account Custom Field
Deal win percent
Back to Account Fields

Custom Field Definition Detail

[Edit](#) [Set Field-Level Security](#) [View Field Accessibility](#) [Where is this used?](#)

Field Information

Field Label	Deal win percent	Object Name	Account
Field Name	Deal_win_percent		
API Name	Deal_win_percent_c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Korapala Sushma, 6/9/2022, 9:03 AM	Modified By	Korapala Sushma, 6/9/2022, 9:03 AM

Formula Options

Data Type	Formula
Decimal Places	2
(Number_of_won_deals_c/Number_of_deals_c)	

Setup > Object Manager

Account

Details

Fields & Relationships

- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Search Layouts
- List View Button Layout
- Hierarchy Columns

Account Validation Rule
Back to Account Validation Rules

Validation Rule Detail

[Edit](#) [Clone](#)

Rule Name	US_Address	Active	✓
Error Condition Formula	OR(AND(LEN(BillingState) > 2, NOT(CONTAINS("AL,AK,AZ,AR,CA,CO,CT,DE,DC,FL,GA,HI,IL,IN,IA,KS,KY,LA,ME,MD,MA,MI,IN,MS,MO,MT,NE,NV,NH,NJ,NM,NY,NC,ND,OH,OK,OR,PA,RI,SC,SD", BillingState)),)AND(LEN(ShippingState) > 2, NOT(CONTAINS("AL,AK,AZ,AR,CA,CO,CT,DE,DC,FL,GA,HI,IL,IN,IA,KS,KY,LA,ME,MD,MA,MI,IN,MS,MO,MT,NE,NV,NH,NJ,NM,NY,NC,ND,OH,OK,OR,PA,RI,SC,SD", ShippingState)),)NOT(OR(BillingCountry = "US", BillingCountry = "USA", BillingCountry = "United States", ISBLANK(BillingCountry))), NOT(OR(ShippingCountry = "US", ShippingCountry = "USA", ShippingCountry = "United States", ISBLANK(ShippingCountry))))		
Error Message	You can not save a new account unless the shipping and billing state fields are valid US state abbreviations, and the country field is either blank or US, USA, or United States.	Error Location	Top of Page
Description			
Created By	Korapala Sushma, 6/9/2022, 9:09 AM	Modified By	Korapala Sushma, 6/9/2022, 9:09 AM

The screenshot shows the Salesforce Object Manager interface for the Account object. On the left, a sidebar lists various configuration options: Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, and Hierarchy Columns. The main content area is titled "Account Validation Rule" and displays the "Validation Rule Detail" for a rule named "Name_Change". The rule is active and has the formula ISCHANGED(Name) && (OR(ISPICKVAL(Type , 'Customer - Direct'), ISPICKVAL(Type , 'Customer - Channel'))). The error message is "You can't change the Account name for "Customer-Direct" or "Customer-Channel"". The description is empty. The created and modified by fields both show "Korapala Sushma, 6/9/2022, 9:11 AM". There are "Edit" and "Clone" buttons at the bottom.

The screenshot shows the Salesforce Object Manager interface for the Account object. On the left, a sidebar lists various configuration options: Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, and Hierarchy Columns. The main content area is titled "Call for Service" and displays the "Custom Field Definition Detail" for a field named "Call_for_Service". The field label is "Call for Service", the field name is "Call_for_Service", and the API name is "Call_for_Service__c". The help text is empty, and the data owner is empty. The field usage is empty. The data sensitivity level and compliance categorization are also empty. The created and modified by fields both show "Korapala Sushma, 6/9/2022, 9:07 AM". The data type is "Formula" with the formula IF(DATE(YEAR(Last_won_deal_date__c)+2,MONTH(Last_won_deal_date__c),DAY(Last_won_deal_date__c))<=TODAY(),"Yes","No"). There are "Edit", "Set Field-Level Security", "View Field Accessibility", and "Where is this used?" buttons at the top of the detail page.

Create Robot Setup Object

The screenshot shows the Salesforce Setup interface with the following details:

Robot Setup Object Details:

- Description:** [Empty]
- API Name:** Robot_Setup_c
- Custom:** ✓
 - Singular Label:** Robot Setup
 - Plural Label:** Robot Setup
- Enable Reports:** [Empty]
- Track Activities:** [Empty]
- Track Field History:** [Empty]
- Deployment Status:** Deployed
- Help Settings:** Standard salesforce.com Help Window

Action Buttons: Edit, Delete

The screenshot shows the Salesforce Setup interface with the following details:

Robot Setup Object Custom Field Definition:

Date

Custom Field Definition Detail:

- Field Information:**

Field Label	Date	Object Name	Robot_Setup
Field Name	Date	Data Type	Date
API Name	Date_c		
Description	Help Text		
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
- General Options:**
 - Required:
 - Default Value: [Empty]
- Validation Rules:** [New] Validation Rules Help ?

Setup Home Object Manager

Robot Setup

Robot Setup Custom Field Notes

Custom Field Definition Detail

Field Information

Field Label	Notes
Field Name	Notes
API Name	Notes__c
Description	
Help Text	
Data Owner	
Field Usage	
Data Sensitivity Level	
Compliance Categorization	

Object Name: Robot_Setup
Data Type: Text

Created By: Korapala_Sushma, 6/9/2022, 9:25 AM Modified By: Korapala_Sushma, 6/9/2022, 9:25 AM

General Options

Required	<input type="checkbox"/>
Unique	<input type="checkbox"/>
Case Sensitive	<input type="checkbox"/>
External ID	<input type="checkbox"/>
Default Value	

Help for this Page

Validation Rules [0]

Fields & Relationships

- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Restriction Rules
- Scoping Rules
- Triggers

Setup Home Object Manager

Robot Setup

Robot Setup Custom Field Notes

Custom Field Definition Detail

Field Information

Field Name	Notes
API Name	Notes__c
Description	
Help Text	
Data Owner	
Field Usage	
Data Sensitivity Level	
Compliance Categorization	

Data Type: Text

Created By: Korapala_Sushma, 6/9/2022, 9:25 AM Modified By: Korapala_Sushma, 6/9/2022, 9:25 AM

General Options

Required	<input type="checkbox"/>
Unique	<input type="checkbox"/>
Case Sensitive	<input type="checkbox"/>
External ID	<input type="checkbox"/>
Default Value	

Text Options

Length	255
--------	-----

Validation Rules

New Validation Rules Help ?

No validation rules defined.

Back To Top Always show me ▾ more records per related list

Fields & Relationships

- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Restriction Rules
- Scoping Rules
- Triggers

The screenshot shows the Salesforce Setup interface with the following details:

Object Manager > **Robot Setup**

Custom Field Definition Detail for **Day of the week**

Field Information

- Field Label: Day of the week
- Field Name: Day_of_the_week
- API Name: Day_of_the_week_c
- Description: (empty)
- Help Text: (empty)
- Data Owner: (empty)
- Field Usage: (empty)
- Data Sensitivity Level: (empty)
- Compliance Categorization: (empty)

Created By: Korapala Sushma, 6/9/2022, 9:32 AM **Modified By**: Korapala Sushma, 6/9/2022, 9:32 AM

Formula Options

Data Type: Formula

```
Case(WEEKDAY(Date__c),  
1,"Sunday",  
2,"Monday",  
3,"Tuesday",  
4,"Wednesday",  
5,"Thursday",  
6,"Friday",  
7,"Saturday",  
Text(WEEKDAY(Date__c)))
```

The screenshot shows the Salesforce Setup interface with the following details:

Object Manager > **Robot Setup**

Custom Field Definition Detail for **Day of the week**

Field Information

- Field Label: Day of the week
- Field Name: Day_of_the_week
- API Name: Day_of_the_week_c
- Description: (empty)
- Help Text: (empty)
- Data Owner: (empty)
- Field Usage: (empty)
- Data Sensitivity Level: (empty)
- Compliance Categorization: (empty)

Created By: Korapala Sushma, 6/9/2022, 9:32 AM **Modified By**: Korapala Sushma, 6/9/2022, 9:32 AM

Formula Options

Data Type: Formula

```
Case(WEEKDAY(Date__c),  
1,"Sunday",  
2,"Monday",  
3,"Tuesday",  
4,"Wednesday",  
5,"Thursday",  
6,"Friday",  
7,"Saturday",  
Text(WEEKDAY(Date__c)))
```

The screenshot shows the Salesforce Setup interface with the following details:

Object Manager > **Robot Setup** > **Opportunity**

Custom Field Definition Detail

Field Information

Field Label	Opportunity	Object Name	Robot_Setup
Field Name	Opportunity	Data Type	Master-Detail
API Name	Opportunity__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Korapala_Sushma, 6/9/2022, 9:33 AM	Modified By	Korapala_Sushma, 6/9/2022, 9:33 AM

Master-Detail Options

Related To	Opportunity	Child Relationship Name	Robot_Setup
Related List Label	Robot Setup		
Sharing Setting	Read/Write: Allows users with at least Read/Write access to the Master record to create, edit, or delete related Detail records.		
Reparentable Master Detail	<input type="checkbox"/>		

Fields & Relationships

- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Restriction Rules
- Scoping Rules
- Triggers

Create Sales Process and Validate Opportunities

The screenshot shows the Salesforce Setup interface with the following details:

Object Manager > **Opportunity**

Custom Field Definition Detail

Field Information

Field Label	Approved	Object Name	Opportunity
Field Name	Approved	Data Type	Checkbox
API Name	Approved__c		
Description			
Help Text			
Data Owner			
Field Usage			
Data Sensitivity Level			
Compliance Categorization			
Created By	Korapala_Sushma, 6/9/2022, 9:36 AM	Modified By	Korapala_Sushma, 6/9/2022, 9:36 AM

General Options

Default Value	Unchecked
---------------	-----------

Field Dependencies

New	Field Dependencies Help ?
-----	---------------------------

Fields & Relationships

- Page Layouts
- Lightning Record Pages
- Buttons, Links, and Actions
- Compact Layouts
- Field Sets
- Object Limits
- Record Types
- Related Lookup Filters
- Search Layouts
- List View Button Layout
- Scoping Rules

Setup Home Object Manager

lead

Feature Settings

Marketing

Lead Assignment Rules

Lead Auto-Response Rules

Lead Processes

Lead Settings

LinkedIn Lead Gen

Lead Gen Fields

LinkedIn Accounts

Web-to-Lead

Sales

Social Accounts and Contacts

Settings

Didn't find what you're looking for?
Try using Global Search.

SETUP Sales Processes

RB Robotics Sales Process

Select a stage from the Available Values list and add it to the Selected Values list to include it in the sales process. Note that removing a stage from the picklist does not remove it from any existing records.

Opportunity Stages

Sales Process: RB Robotics Sales Process

Namespace Prefix:

Description:

Available Values

Selected Values

Add Remove

Help for this Page

Setup Home Object Manager

SETUP > OBJECT MANAGER Opportunity

Details

Fields & Relationships

Page Layouts

Lightning Record Pages

Buttons, Links, and Actions

Compact Layouts

Field Sets

Object Limits

Record Types

Related Lookup Filters

Search Layouts

List View Button Layout

Scoping Rules

Validation Rule Detail

Rule Name: RB_High_Value_Opp

Error Condition Formula: IF((Amount > 100000 && Approved__c <> True && ISPICKVAL(StageName,'Closed Won')),True,False)

Error Message: Appropriate value should be given

Description:

Created By: Korapala Sushma, 6/9/2022, 9:45 AM

Active: ✓

Error Location: Top of Page

Modified By: Korapala Sushma, 6/9/2022, 9:45 AM

Edit Clone

Help for this Page

Automate Opportunities

The screenshot shows the Salesforce Setup interface with the Approval Processes page open. The process is named "Prospect Approval".

Process Definition Detail:

- Unique Name: Prospect_Approval
- Description: Opportunity: Prospect Approval
- Entry Criteria: (Opportunity: Stage EQUALS Negotiation/Review) AND (Opportunity: Amount GREATER THAN 100000)
- Record Editability: Administrator ONLY
- Next Automated Approver Determined By: Manager of Record Submitter
- Approval Assignment Email Template: Sales_Opportunity_Approval_Status_Email
- Initial Submitters: User_Nushi_Davoud, Opportunity Owner
- Created By: Korapala_Sushma, 6/9/2022, 10:47 AM
- Modified By: Korapala_Sushma, 6/9/2022, 11:20 AM

Initial Submission Actions:

- Action: Record Lock, Description: Lock the record from being edited
- Action: Field Update, Description: Approval

Approval Steps:

Action	Step Number	Name	Description	Criteria	Assigned Approver	Reject Behavior
Show Actions	Edit	1	Approval for Prospect		User_Nushi_Davoud	Final Rejection

Final Approval Actions:

- Action: Record Lock, Description: Lock the record from being edited
- Action: Field Update, Description: Stage_Closed_Won
- Action: Email Alert, Description: Sales_Opportunity_Approved_Request_Mail
- Action: Field Update, Description: Approved_Check

Final Rejection Actions:

- Action: Record Lock, Description: Unlock the record for editing
- Action: Field Update, Description: Stage_Neg
- Action: Email Alert, Description: Sales_Approval_Email

Recall Actions:

- Action: Record Lock, Description: Unlock the record for editing
- Action: Email Alert, Description: Finance_Account_Creation
- Action: Field Update, Description: StageClosedWon
- Action: Field Update, Description: approvedcheck

Process Automation Specialist | Lightning Experience | Salesforce | Field Updates | Salesforce | Student Dashboard | SI-34767-1656054334 | SmartInternz - Your temporary...

Field Update Detail

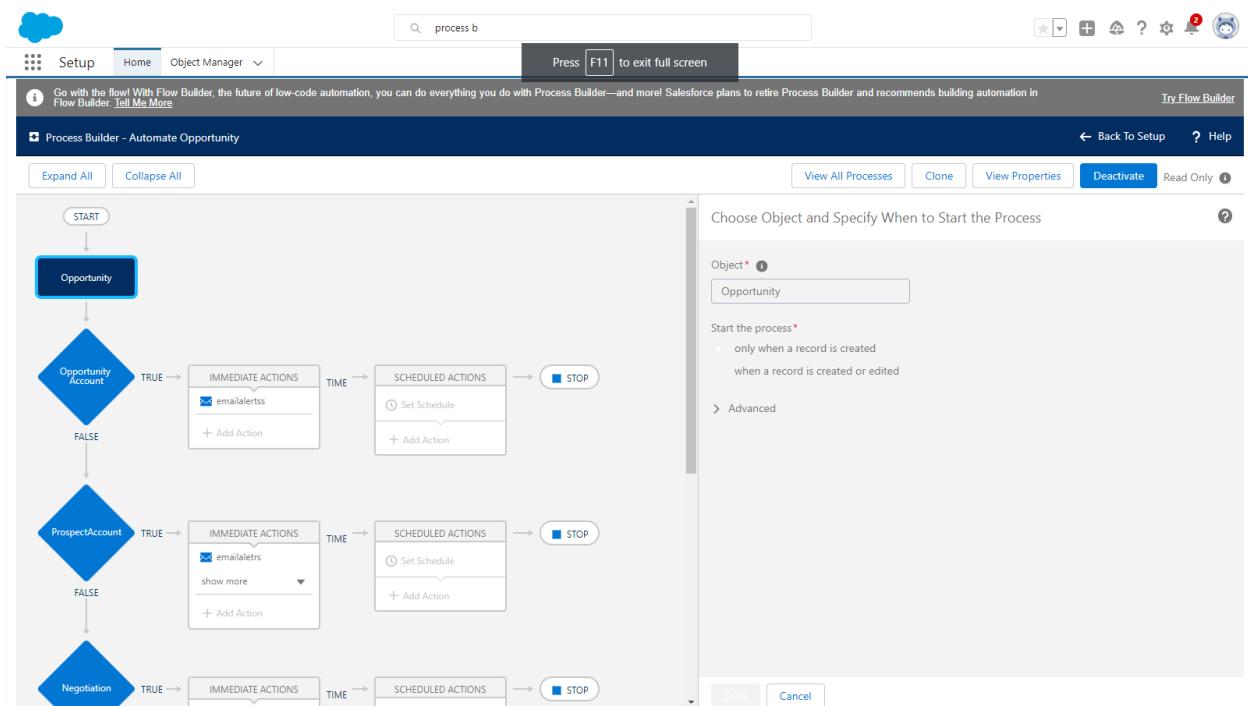
- Name: Approval
- Unique Name: Approval
- Description: Opportunity
- Object: Opportunity
- Field to Update: Opportunity: Stage
- Field Data Type: Picklist
- Re-evaluate Workflow Rules after Field Change:
- New Field Value: Awaiting Approval

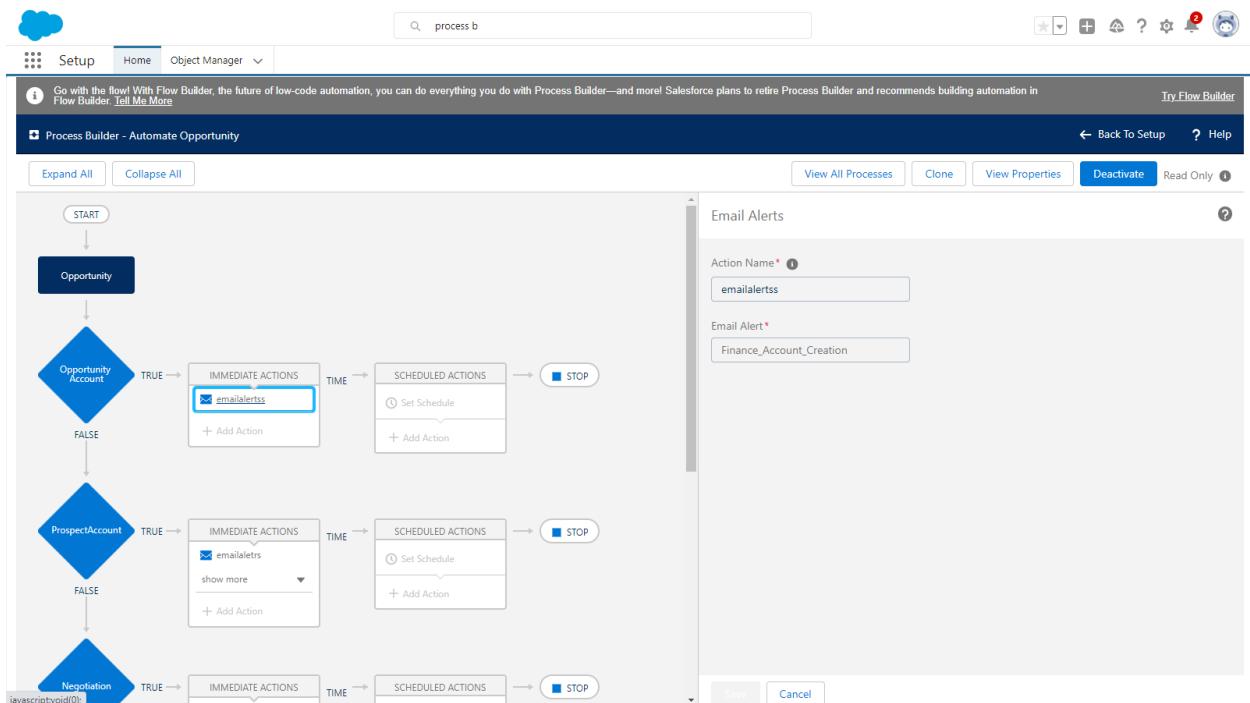
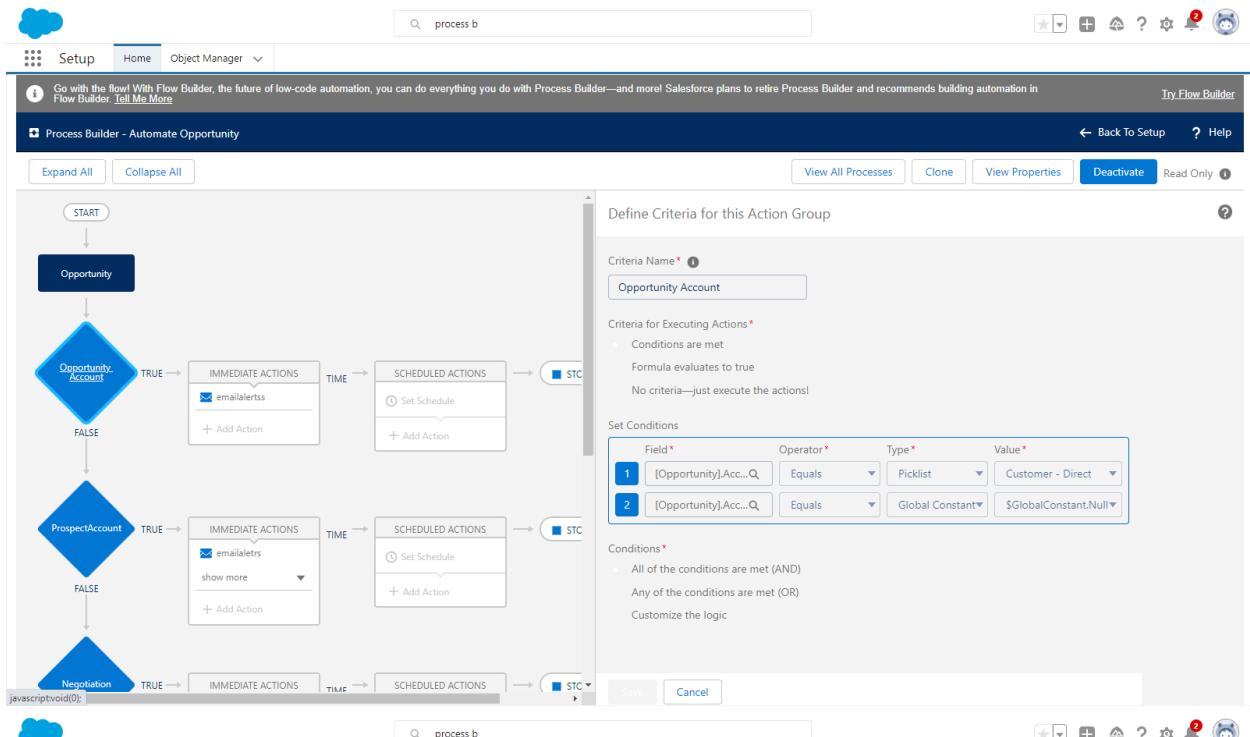
Rules Using This Field Update
This field update is currently not used by any rules.

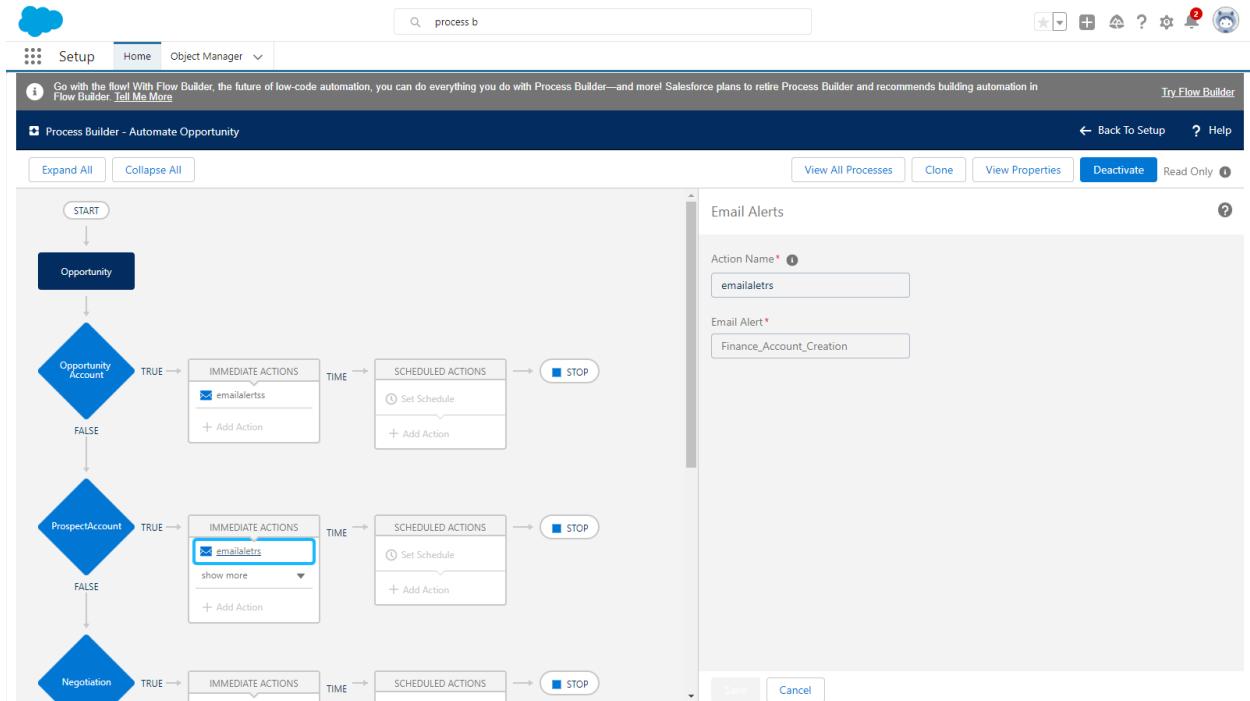
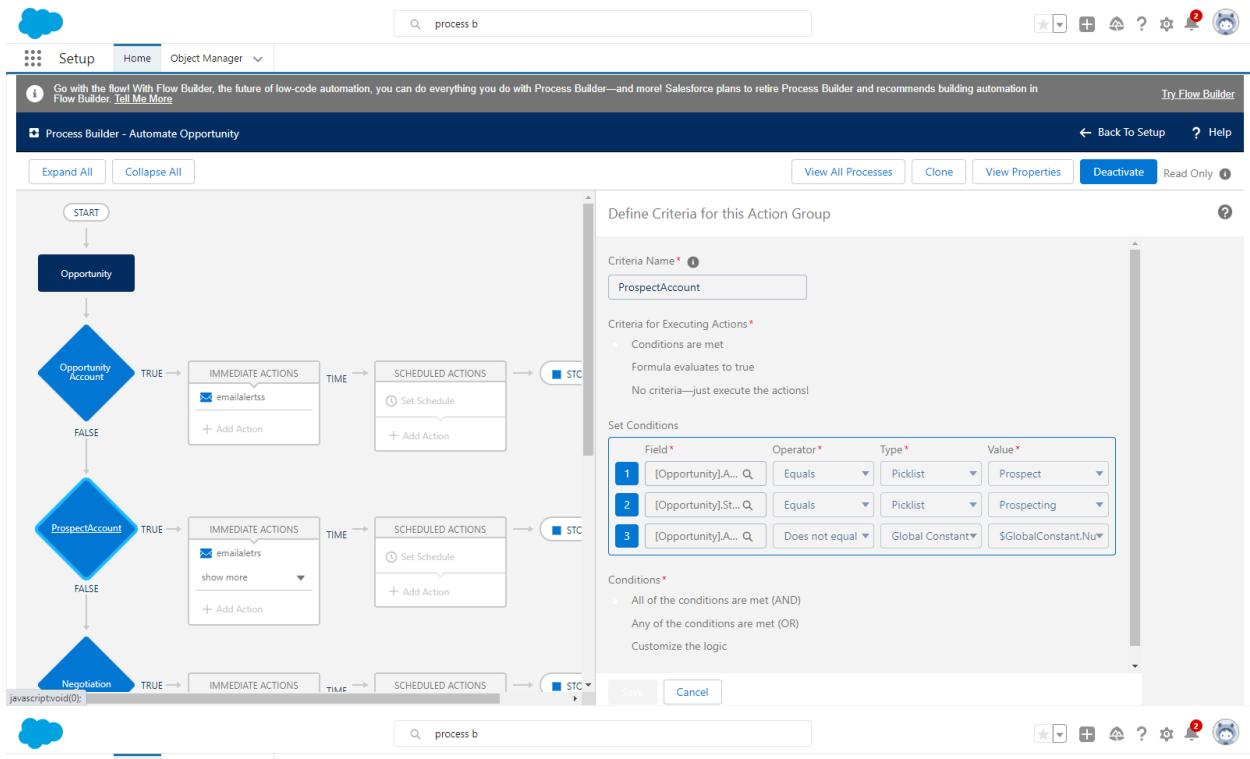
Approval Processes Using This Field Update

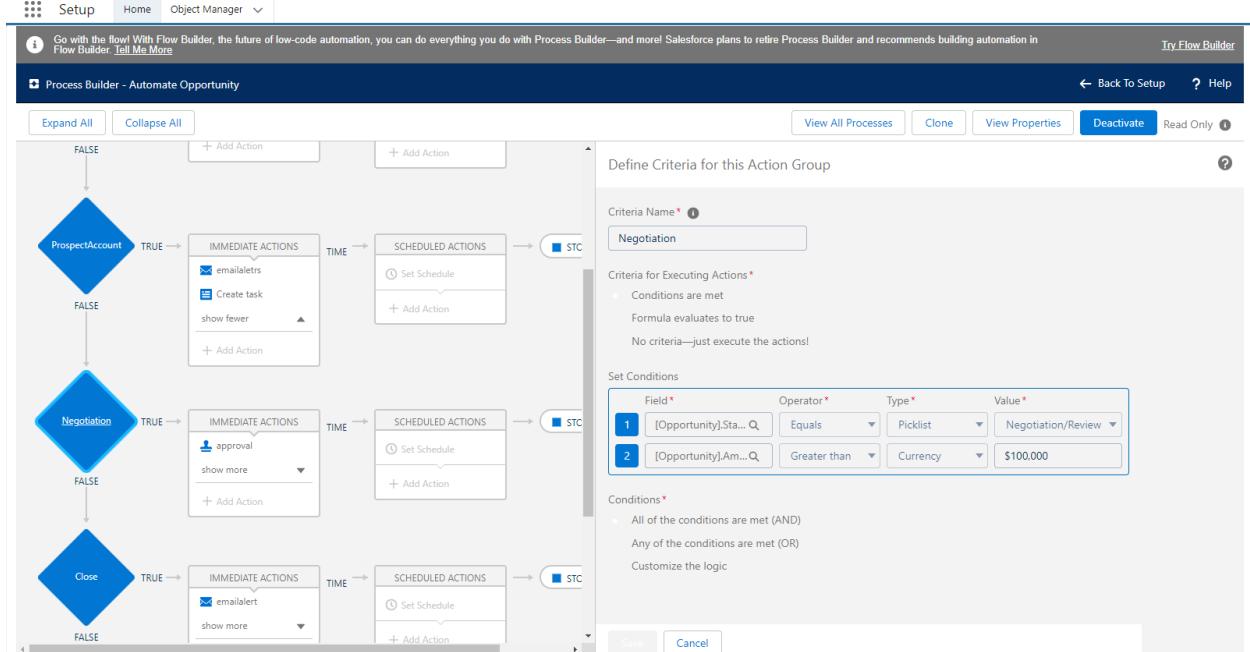
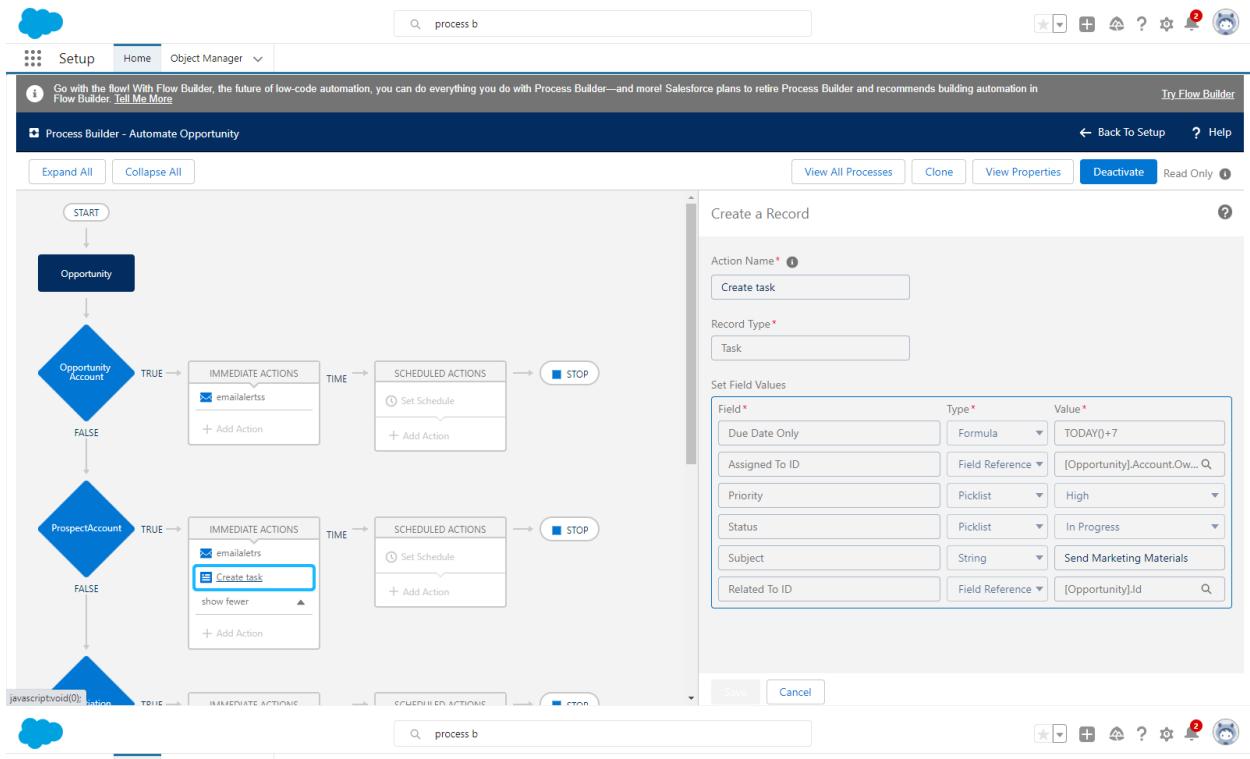
Action	Approval Process Name	Description	Type	State
Edit	Prospect Approval		Opportunity	Active

Entitlement Processes Using This Field Update
This field update is currently not used by any entitlement processes.







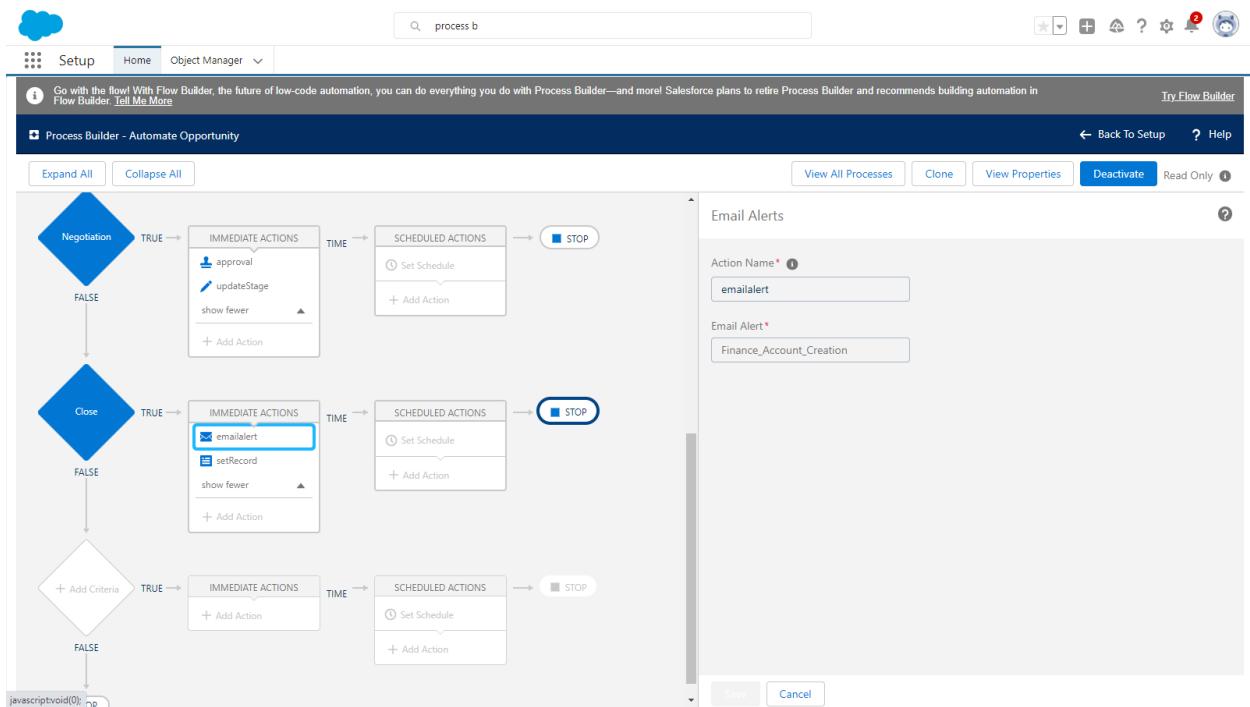
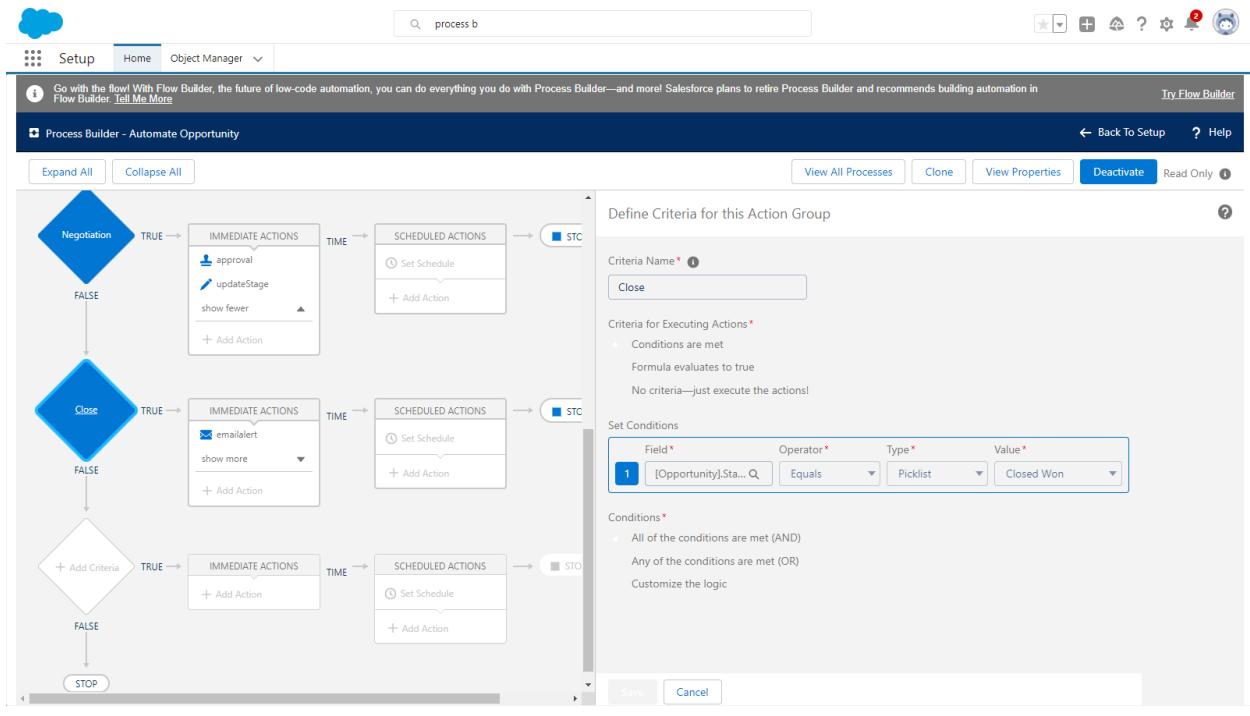


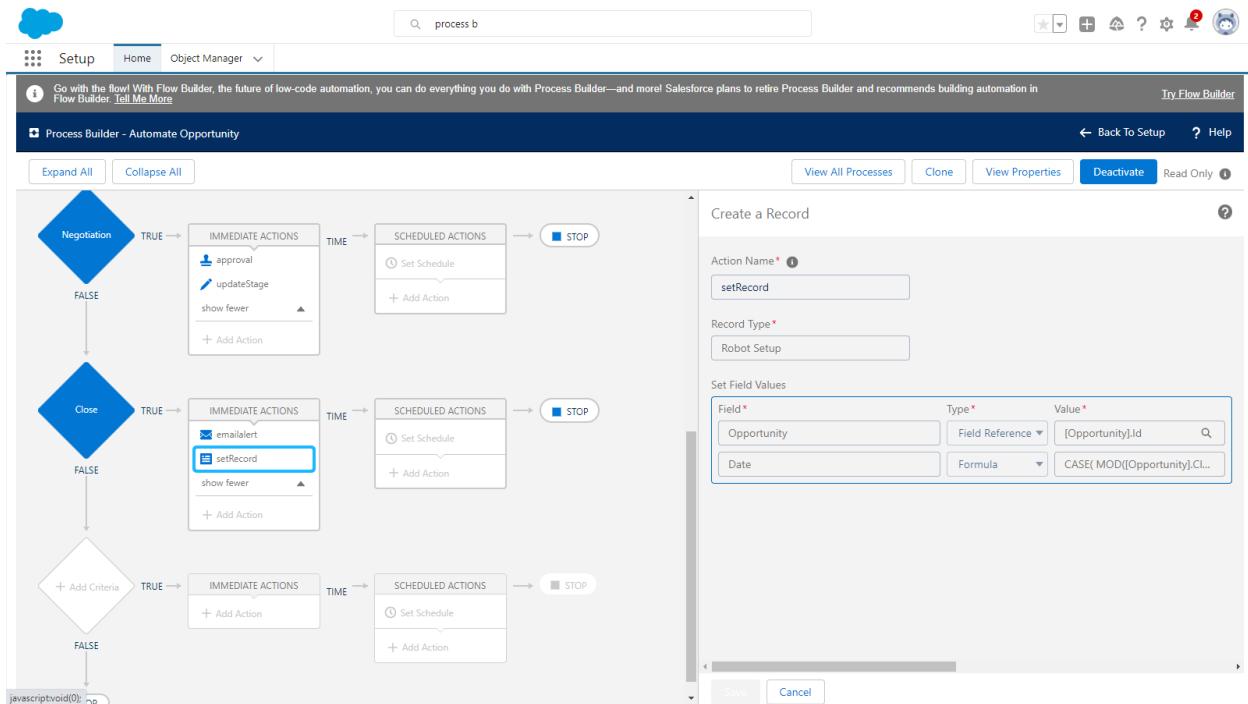
The screenshot shows the Salesforce Process Builder interface with a process titled "Process Builder - Automate Opportunity". The process starts with a decision diamond "ProspectAccount". If FALSE, it ends. If TRUE, it triggers immediate actions: "emailalerts" and "Create task", followed by scheduled actions: "Set Schedule" and a STOP action. This is followed by another decision diamond "Negotiation". If FALSE, it ends. If TRUE, it triggers immediate actions: "approval" (which is highlighted with a blue box) and "updateStage", followed by scheduled actions: "Set Schedule" and a STOP action. Finally, a third decision diamond "Close" is reached. If FALSE, it ends. If TRUE, it triggers immediate actions: "emailalert" and "show more", followed by scheduled actions: "Set Schedule" and a STOP action. The process ends with a "Cancel" button.

The screenshot shows the Salesforce Process Builder interface with a process titled "Process Builder - Automate Opportunity". The process flow is as follows:

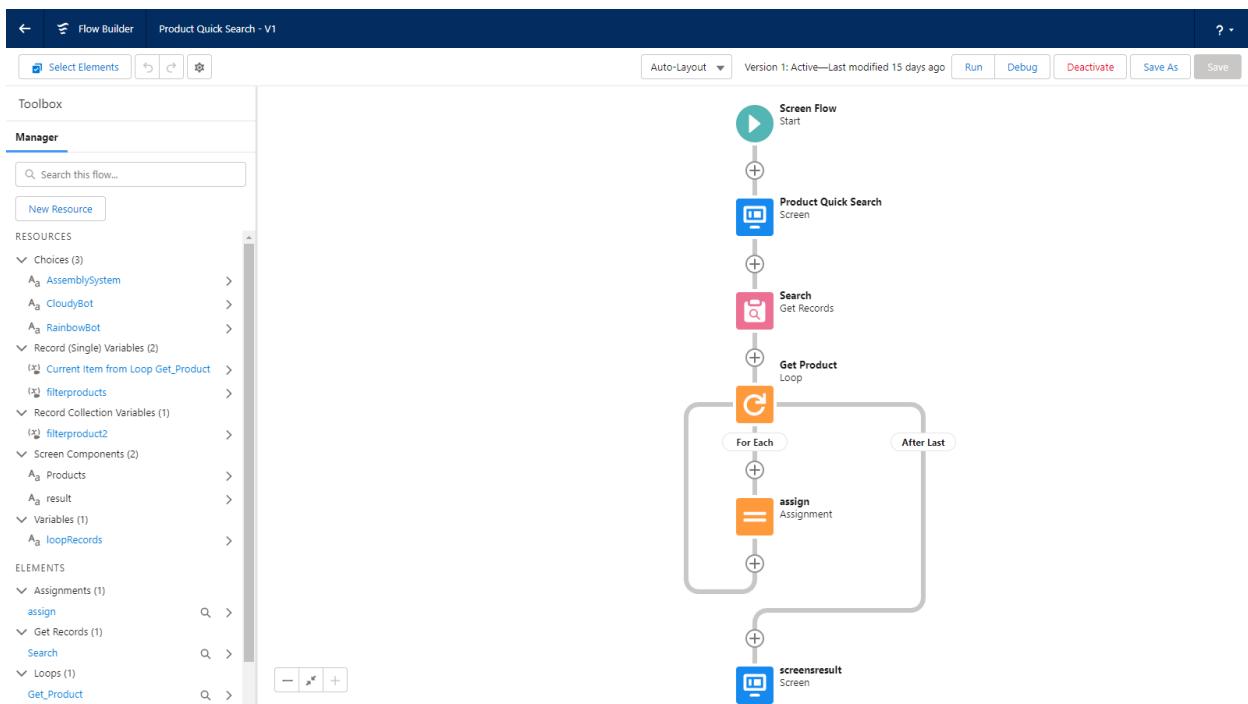
- Decision Point 1: ProspectAccount** (TRUE path)
 - IMMEDIATE ACTIONS: emailalerts, Create task, show fewer
 - TIME → SCHEDULED ACTIONS: Set Schedule, then a STOP action
- Decision Point 2: Negotiation** (TRUE path)
 - IMMEDIATE ACTIONS: approval, updateStage (highlighted in blue), show fewer
 - TIME → SCHEDULED ACTIONS: Set Schedule, then a STOP action
- Decision Point 3: Close** (TRUE path)
 - IMMEDIATE ACTIONS: emailalert, show more
 - TIME → SCHEDULED ACTIONS: Set Schedule, then a STOP action

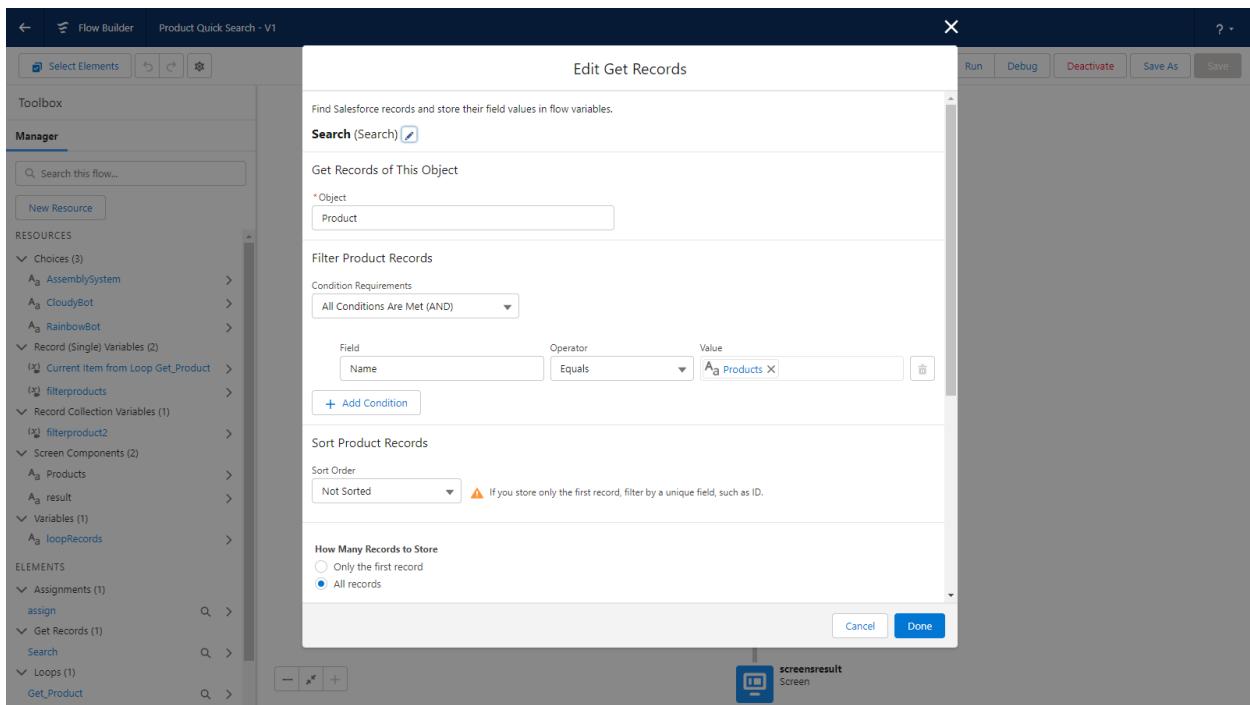
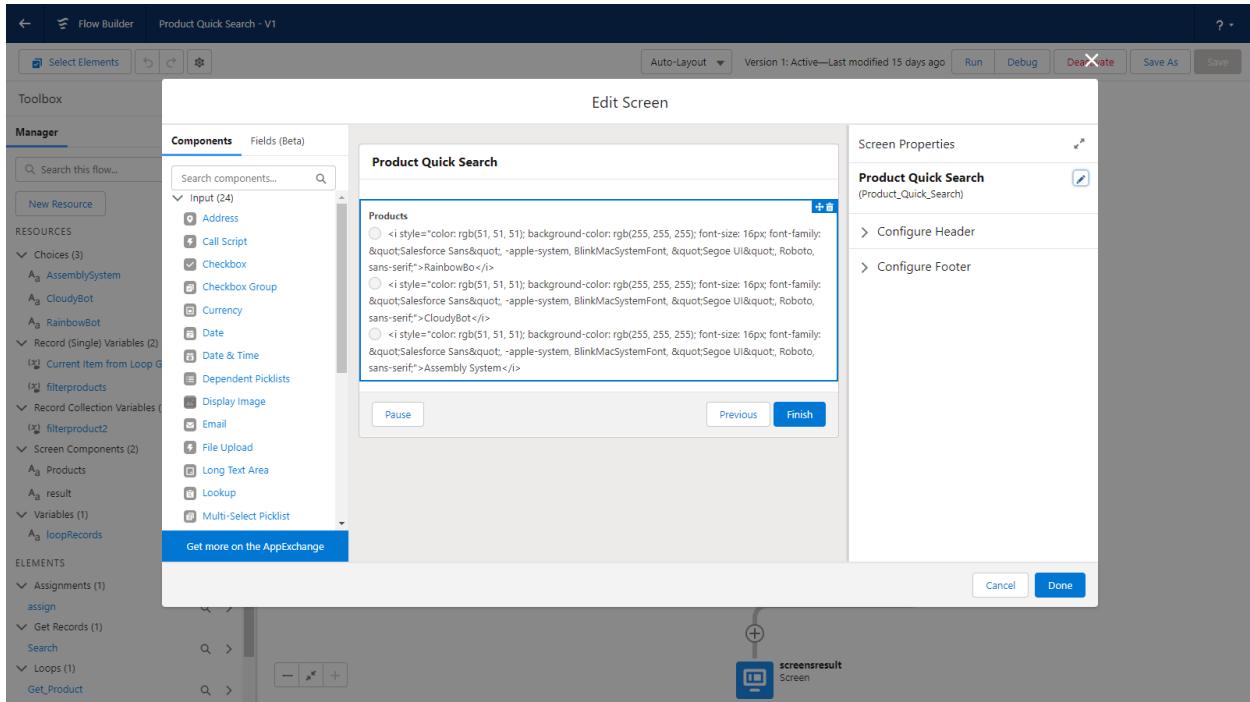
A sidebar on the right shows the configuration for the "updateStage" action under "Update Records". It includes fields for Action Name (updateStage), Record ([Opportunity]), Criteria for Updating Records (Updated records meet all conditions, No criteria—just update the records!), and Field Values (Stage: Picklist, Value: Awaiting Approval).

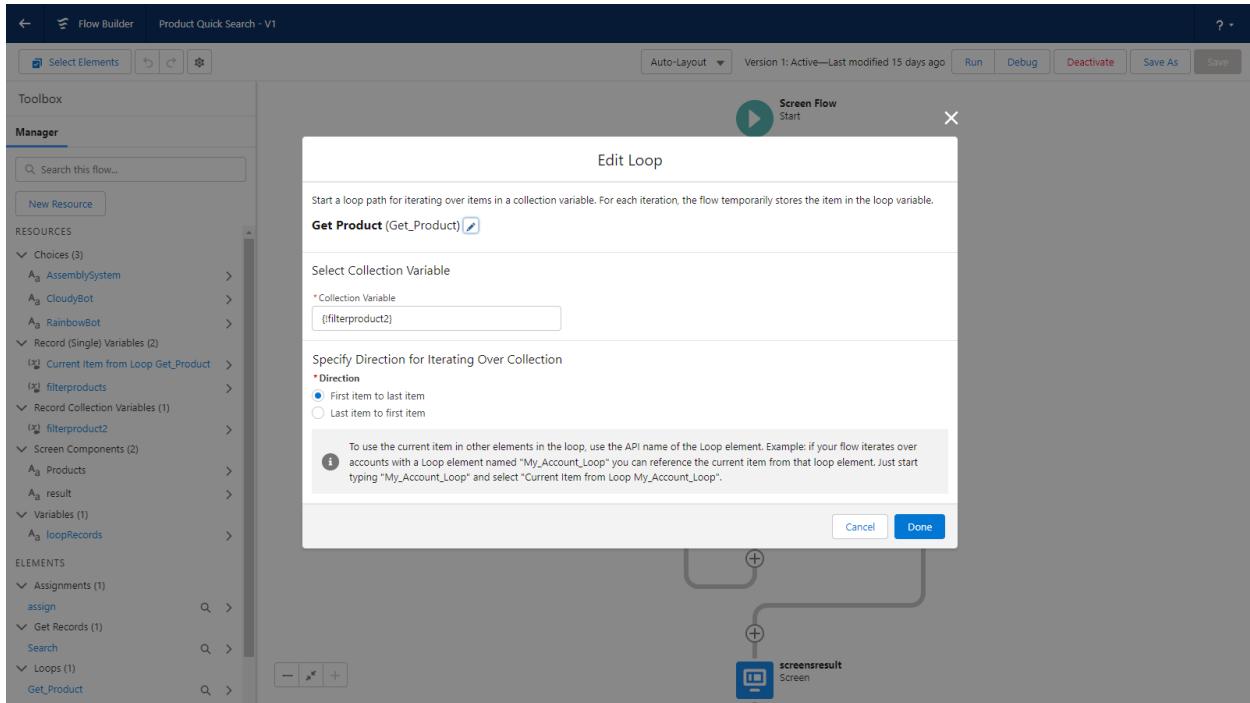
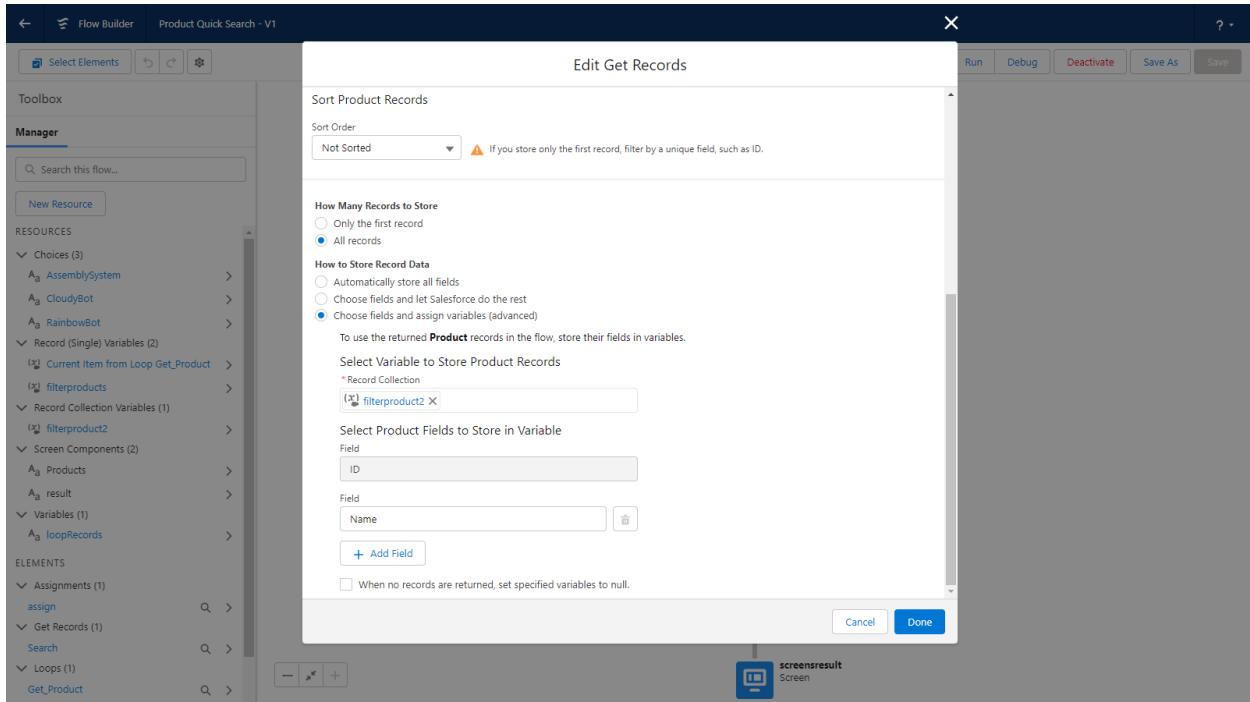


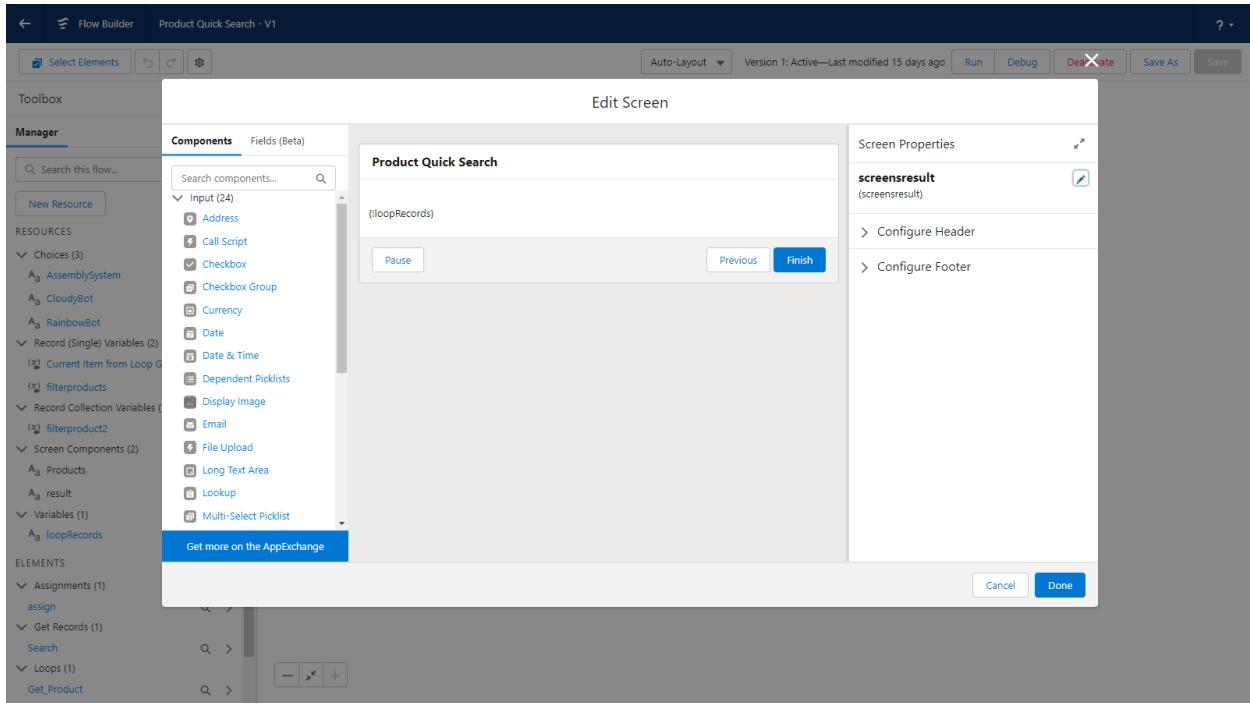
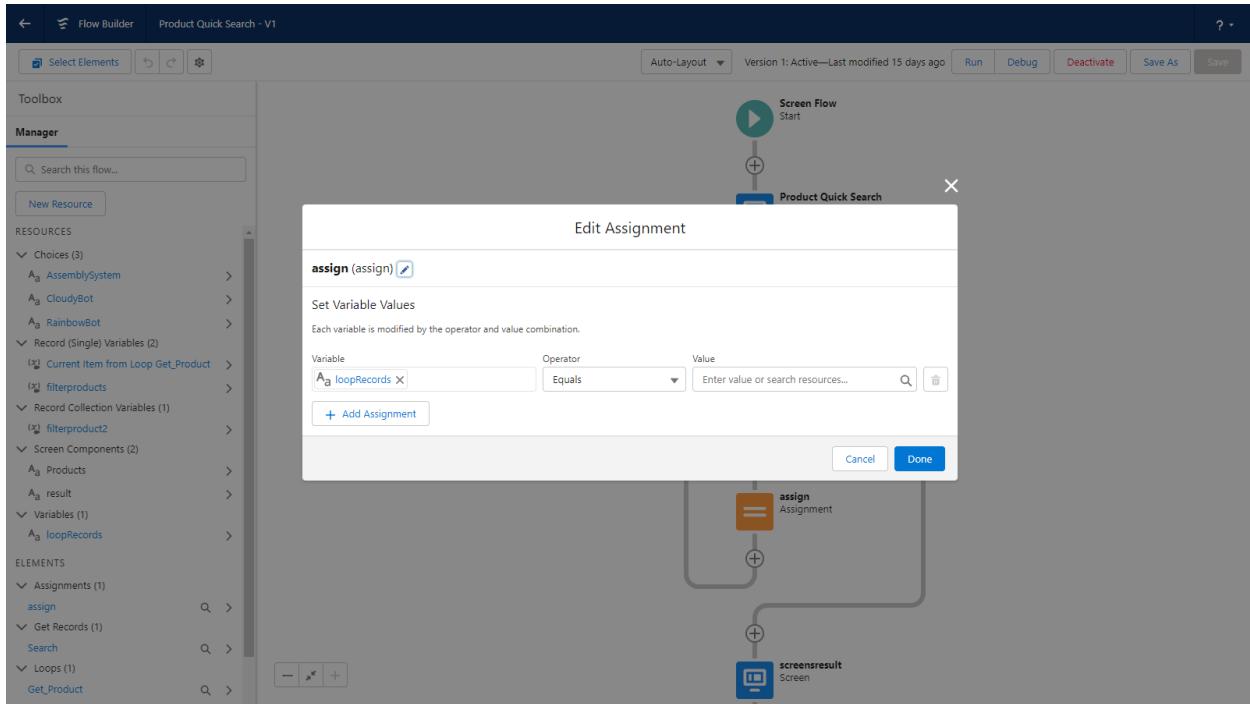


Create Flow for Opportunities and Automate Setups









The screenshot shows the Salesforce Setup interface with the following details:

- Header:** Includes the Salesforce logo, "SETUP > OBJECT MANAGER", "Opportunity", a search bar ("Search Setup"), and a message "Press F11 to exit full screen".
- Left Sidebar:** A navigation menu under "Lightning Record Pages" with the following items:
 - Details
 - Fields & Relationships
 - Page Layouts
 - Lightning Record Pages** (selected)
 - Buttons, Links, and Actions
 - Compact Layouts
 - Field Sets
 - Object Limits
 - Record Types
 - Related Lookup Filters
 - Search Layouts
 - List View Button Layout
 - Scoping Rules
 - Triggers
 - Flow Triggers
 - Validation Rules
- Content Area:** Displays the "Opportunity_Lightning_Page" record.
 - Lightning Page Detail:** Shows the page's name ("Opportunity_Lightning_Page") and label ("Opportunity Lightning Page").
 - Information:** A table with columns "Name", "Description", and "Label".
 - Assignments By App:** A section showing "No Assignments to display".
 - Assignments By App, Record Type, and Profile:** Another section showing "No Assignments to display".