

Steps to extend Action Plans for custom Object

This quick guide will help you through extending Action Plans application to custom objects. This guide is intended for users lacking of a programmer background, but programmers will find this guide useful as well.

We will name this new Objects MyCustomObject.

Check if there's any consideration with Chatter set up

Adding 'Create Action Plans' Button to custom objects list

Create a custom button on your Object

Edit TestObject Custom Button or Link

Create TestObject Action Plan

Custom Button or Link Edit [Save] [Quick Save] [Preview] [Cancel]

Label: Create TestObject Action Plan
Name: Create_TestObject_Action_Plan [i]
Description:
Display Type:
☐ Detail Page Link [View example](#)
☐ Detail Page Button [View example](#)
☒ List Button [View example](#)
☒ Display Checkboxes (for Multi-Record Selection)
Behavior: Execute JavaScript [v] [View Behavior Options](#)
Content Source: OnClick JavaScript [v]

Select Field Type: TestObject [v] Insert Field: - Insert Merge Field - [v] Insert Operator: [v] Functions: - All Function Categories - [v]
ABS
AND
BECOME
CASE
CEILING
Insert Selected Function

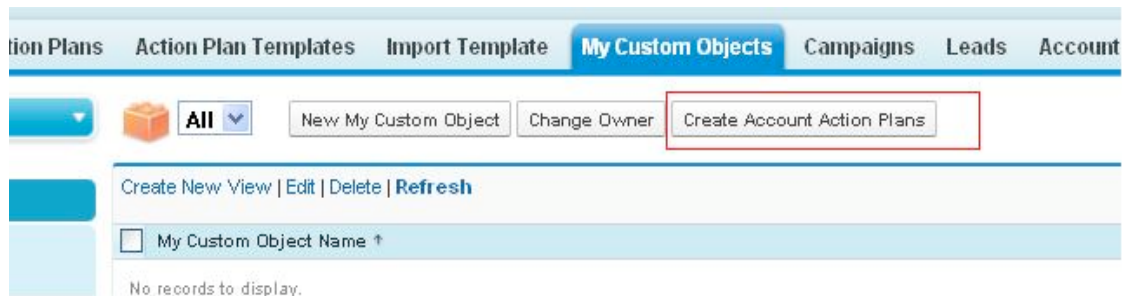
```
var records = (GETRECORDIDS($ObjectType,Lead));  
var oName = "MyCustomObject_c";  
var error = false;  
var msg = "";  
if( records.length > 40 ) {  
    error = true;  
    msg = "{!$Label.ap_Errors_moreThan20records}";  
}  
if( records.length <= 1 ) {  
    error = true;  
    msg = "{!$Label.ap_Errors_selectOneRecord}";  
}  
if( error ) {  
    alert(msg);  
} else {  
    location.href = Yapex/ActionPlanCreation?refId='+records+'&refType='+oName;  
}
```

var oName contains the name of the custom object

this logic controls that the number of records selected is between 1 and 40

This button needs to be added to the search layout of your custom object :

On the definition Page of your Custom object under 'Search Layouts' section, edit 'My Custom Objects List View '. Add the custom button just created (listed under “Available buttons”) to “Selected buttons”.



Adding 'Create Action Plan' button to object detail page

You need to add a custom ' Lookup Relationship' from ActionPlan's Object to your Custom Object.

Code Modifications

To indicate the changes that had to be made in the code of the app in order to extend it for new objects we created a list pointing out all the changes that must be made.

Each item of the list will consist in : the filename , the classname , the method name and the line number for each change. It will include examples of the code before and after the changes.

Classes

Change 1 :

File : ActionPlanCreationController.cls

Class: ActionPlanCreationController

method : initEditActionPlan

In the query you need to add the API name for the lookUp relationship with you new object

code before:

```
this.actionPlan = [
    select a.Id, a.Name, a.CreatedById, a.StartDate__c,
           a.Action_Plan_Template__r.Name, a.Action_Plan_Template__r.Id,
           a.Action_Plan_Template__c, a.Account__c, a.Contact__c,
           a.Lead__c, a.Opportunity__c, a.Contract__c, a.Case__c,
           a.Campaign__c , a.SkipWeekends__c, a.SkipDay__c
    from ActionPlan__c a
           Where a.Id =: this.actionPlanId limit 1 ];
```

code after:

```
this.actionPlan = [
    select a.Id, a.Name, a.CreatedById, a.StartDate__c,
           a.Action_Plan_Template__r.Name, a.Action_Plan_Template__r.Id,
           a.Action_Plan_Template__c, a.Account__c, a.Contact__c,
           a.Lead__c, a.Opportunity__c, a.Contract__c, a.Case__c,
           a.Campaign__c , a.SkipWeekends__c, a.SkipDay__c,
           a.My_Custom_Object__c
    from ActionPlan__c a
           Where a.Id =: this.actionPlanId limit 1 ];
```

Change 2 :

File : ActionPlanCreationController.cls

Class: ActionPlanCreationController

method : retrieveNameRelatedObject

Add a conditional statement at the end of the list of conditionals.

The value in 'CustomObject' must much the value you enter when defining the logic for the Create Actions Plans button in out case :

var oName = 'MyCustomObject__c';

code after:

```
else if( 'MyCustomObject'.equalsIgnoreCase( relatedObjectName ) ){
    this.relatedObjName = 'MyCustomObject__c';
    for( My_Custom_Object__c custObj :
        [ Select Name from My_Custom_Object__c where Id IN : relatedObjIds ] ){
        this.relatedObjNameList.add( custObj.Name );
    }
}
```

Change 3 :

File : ActionPlanDetailController.cls

Class: ActionPlanDetailController

method : ActionPlanDetailController constructor

code before:

```
this.actionPlan = [ select a.Id, a.Name, a.CreatedById, a.StartDate__c,
    a.Action_Plan_Template__r.Name, a.Action_Plan_Template__r.Id,
    a.Action_Plan_Template__c, a.Account__c, a.Contact__c, a.Lead__c,
    a.Opportunity__c, a.Contract__c, a.Case__c, a.Campaign__c ,
    a.SkipWeekends__c, a.SkipDay__c, a.Owner.Name, a.CreatedDate,
    a.LastModifiedById, a.LastModifiedDate
    from ActionPlan__c a
    Where a.Id =: this.actionPlanId limit 1 ];
```

code after:

```
this.actionPlan = [ select a.Id, a.Name, a.CreatedById, a.StartDate__c,
    a.Action_Plan_Template__r.Name, a.Action_Plan_Template__r.Id,
    a.Action_Plan_Template__c, a.Account__c, a.Contact__c, a.Lead__c,
    a.Opportunity__c, a.Contract__c, a.Case__c, a.Campaign__c ,
    a.SkipWeekends__c, a.SkipDay__c, a.Owner.Name, a.CreatedDate,
    a.LastModifiedById, a.LastModifiedDate, a.My_Custom_Object__c
    from ActionPlan__c a
    Where a.Id =: this.actionPlanId limit 1 ];
```

Change 4 :

File : ActionPlanDetailController.cls

Class: ActionPlanDetailController

method : getRelatedObjectName

Add a conditional statement at the end of the list of conditionals.

code after:

```

else if( this.actionPlan.My_Custom_Object__c != null ){
    My_Custom_Object__c o = [select Name from My_Custom_Object__c
                               where id =: this.actionPlan.My_Custom_Object__c limit 1];
    objName = o.Name;
}

```

Change 5 :

File : ActionPlanDetailController.cls

Class: ActionPlanDetailController

method : checkGetRelatedObjectName test method

Add a block for testing the retrieving the names of the related objects within an Action Plan when those objects have the type My_Custom_Object__c .

Change 6:

File : ActionPlanObjectTriggerTest.cls

Class: ActionPlanObjectTriggerTest

method : testMyCustomObject Trigger

You need to add a test method

Change 7:

File : ActionPlansBatchBuilderUtilities.cls

Class: ActionPlansBatchBuilderUtilities

method : ActionPlansBatchBuilderUtilities

Add a conditional statement at the end of the list of conditionals.

code after:

```

if ( 'MyCustomObject__c'.equalsIgnoreCase( refType ) ){
    ap.My_Custom_Object__c = relObjId;
}

```

Change 7:

File : ActionPlansBatchBuilderUtilities.cls

Class: ActionPlansBatchBuilderUtilities

method : generateActionPlans

Add a conditional statement at the end of the list of conditionals. After this line : //Add here code sample to extend for a custom Object

code after:

```
if( apAux.Contract__c != null ){  
    t.WhatId = apAux.Contract__c;  
}
```

Change 8:

File : ActionPlansTaskTriggerUtilities.cls

Class: ActionPlansTaskTriggerUtilities

method : dependentActionPlansTemplateTask

code before:

```
this.dependentAptts = [select a.Action_Plan__r.SkipWeekends__c,  
    a.Action_Plan__r.Account__c, a.Priority__c, a.Status__c,  
    a.Action_Plan__r.Contact__c, a.Action_Plan__r.Lead__c,  
  
    a.Action_Plan__r.Opportunity__c, a.Action_Plan__r.Contract__c,  
    a.Action_Plan__r.Case__c, a.Action_Plan__r.Campaign__c ,  
    a.Action_Plan__r.StartDate__c, a.Action_Plan__c,  
  
    a.Type__c, a.Subject__c, a.ActivityDate__c, a.Dependent__c,  
    a.Name, a.User__c, a.User__r.Name, a.Id, a.DaysFromStart__c,  
    a.CreatedById, a.APTaskTemplate__r.Subject__c,  
    a.APTaskTemplate__c, a.SendEmail__c, a.APTaskTemplate__r.Name,  
    a.Reminder__c, a.Comments__c, a.Minutes_Reminder__c  
  
        from APTaskTemplate__c a  
    where a.APTaskTemplate__c IN : this.closedTasks ];
```

code after:

```
this.dependentAptts = [select a.Action_Plan__r.SkipWeekends__c,  
    a.Action_Plan__r.Account__c, a.Priority__c, a.Status__c,  
    a.Action_Plan__r.Contact__c, a.Action_Plan__r.Lead__c,  
  
    a.Action_Plan__r.Opportunity__c, a.Action_Plan__r.Contract__c,  
    a.Action_Plan__r.Case__c, a.Action_Plan__r.Campaign__c ,  
    a.Action_Plan__r.StartDate__c, a.Action_Plan__c,  
  
    a.Type__c, a.Subject__c, a.ActivityDate__c, a.Dependent__c,  
    a.Name, a.User__c, a.User__r.Name, a.Id, a.DaysFromStart__c,  
    a.CreatedById, a.APTaskTemplate__r.Subject__c,  
    a.APTaskTemplate__c, a.SendEmail__c, a.APTaskTemplate__r.Name,  
    a.Reminder__c, a.Comments__c, a.Minutes_Reminder__c,  
  
    a.Action_Plan__r.My_Custom_Object__c  
  
        from APTaskTemplate__c a
```

```
where a.APTaskTemplate__c IN : this.closedTasks ];
```

Change 9:

File : ActionPlansTaskTriggerUtilities.cls

Class: ActionPlansTaskTriggerUtilities

method : populateUpdateApttsList

Add a conditional statement at the end of the list of conditionals. After this line : //Add custom object code

code after:

```
if( dependentAptt.Action_Plan__r.My_Custom_Object__c != null ){  
    t2.WhatId = dependentAptt.Action_Plan__r.My_Custom_Object__c;  
}
```

Change 10:

File : ActionPlansTestUtilities.cls

Class: ActionPlansTestUtilities

method : Add methods for code coverage

The following changes all refer to this file : **ActionPlansUtilities.cls** and class **ActionPlansUtilities**

Change 11:

method : getActionPlans

code before:

```
return [    select a.Id, a.Name, a.CreatedById, a.StartDate__c,  
            a.Action_Plan_Template__r.Name, a.Action_Plan_Template__r.Id,  
            a.Action_Plan_Template__c, a.Account__c, a.Contact__c, a.Lead__c,  
            a.Opportunity__c , a.Contract__c, a.Case__c, a.Campaign__c  
            from ActionPlan__c a limit 1000 ];
```

code after:

```
return [    select a.Id, a.Name, a.CreatedById, a.StartDate__c,  
            a.Action_Plan_Template__r.Name, a.Action_Plan_Template__r.Id,  
            a.Action_Plan_Template__c, a.Account__c, a.Contact__c, a.Lead__c,  
            a.Opportunity__c , a.Contract__c, a.Case__c, a.Campaign__c ,  
            a.My_Custom_Object__c  
            from ActionPlan__c a limit 1000 ];
```

Change 12:

method : saveAp

Add a conditional statement at the end of the list of conditionals.

code after:

```
else if( this.actionPlan.My_Custom_Object__c!= null ) {  
    objId = this.actionPlan.My_Custom_Object__c;  
    objName = 'MyCustomObject__c';  
}
```

Change 13:

method : generateObjectFeed

Add a conditional statement at the end of the list of conditionals ..

code after:

```
else if( ap.My_Custom_Object__c != null )      {  
    objId      = ap.My_Custom_Object__c;  
    objName    = 'MyCustomObject';  
    r          = My_Custom_Object__c.sObjectType.getDescribe();  
    chatter_enabled =  r.IsFeedEnabled();  
}
```

Change 14:

method : getRecordOwnerId

Add a conditional statement at the end of the list of conditionals.

code after:

```
else if( ap.My_Custom_Object__c != null )      {  
    objId = ap.My_Custom_Object__c;  
}
```

Change 15:

method : isObjectEnabledFeed

Add a conditional statement at the end of the list of conditionals .

code after:

```
else if( ap.My_Custom_Object__c != null )      {  
    objName = 'MyCustomObject__c';  
}
```

Change 16:

method : hasRelatedObject

Add this condition to the Boolean sentence :

code after:

```
actionPlan.My_Custom_Object__c != null
```

Change 17:

method : getRelatedObjectOptions

Add this sentence before the return sentence :

code after:

```
options.add( new SelectOption( 'MyCustomObject__c',  
                               Schema.Sobjecttype.My_Custom_Object__c.label ) );
```

Change 18:

method : relatedObjectId

Add this sentence at the end of the list of conditional :

code after:

```
else if( actionPlan.My_Custom_Object__c != null ) {  
    currentId = actionPlan.My_Custom_Object__c;  
}
```

Change 19:

File : ActionPlansUtilitiesTest.cls

Class: ActionPlansUtilitiesTest

method : APUtestGenerateObjectFeed

Add a block for generating feeds for your custom object

code after:

```
ap = ge.createNewActionPlan( 'MyCustomObject', 0 );
apu = new ActionPlansUtilities(ap);
apu.generateObjectFeed(ap.Id);
```

Static resources

No changes required

Pages

File : ActionPlanCreation.page

Add the following code at line 170:

```
<apex:outputPanel id="my_Custom_Object__c" rendered="{!relatedObjectSelected ==
'MyCustomObject__c'}" >
    <apex:pageBlockSectionItem id="myCustomObjectLookupSection">
        <apex:inputField id="oName"
            styleClass="{!IF(throwError !=null,'errorBorder','')}"
            value="{!actionPlan.My_Custom_Object__c}" />
        <apex:message for="oName"/>
    </apex:pageBlockSectionItem>
</apex:outputPanel>
```

Add the following code at line 197:

```
<apex:outputField value="{!actionPlan.My_Custom_Object__c}"
    rendered="{!(actionPlan.My_Custom_Object__c != null)}" />
```

File : ActionPlanDetail.page

Add the following code at line 36:

```
<apex:outputField value="{!actionPlan.CustomObject__c}"
    rendered="{!(actionPlan.CustomObject__c != null)}" />
```

Action Plan object:

Add the following condition to 'Reference Object Can Not Be Null' validation rule :

`OR( My_Custom_Object__c == null, My_Custom_Object__c == " )`

Triggers:

you'll need to create a trigger to handle ActionPlan deletion and restoration when your Custom objects is deleted and undeleted.