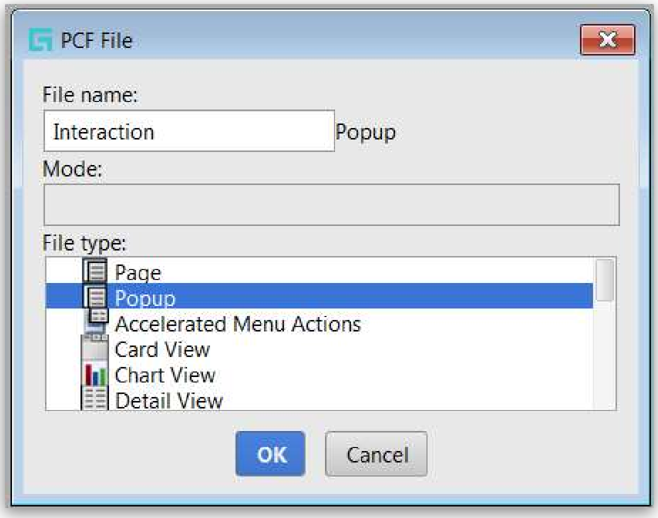
5.3 Lab Solution: Popup locations

Solution

Exact details on how to complete the lab.

1. **Create the traininglabs PCF folder if it doesn't exist**
2. **Create the InteractionPopup**

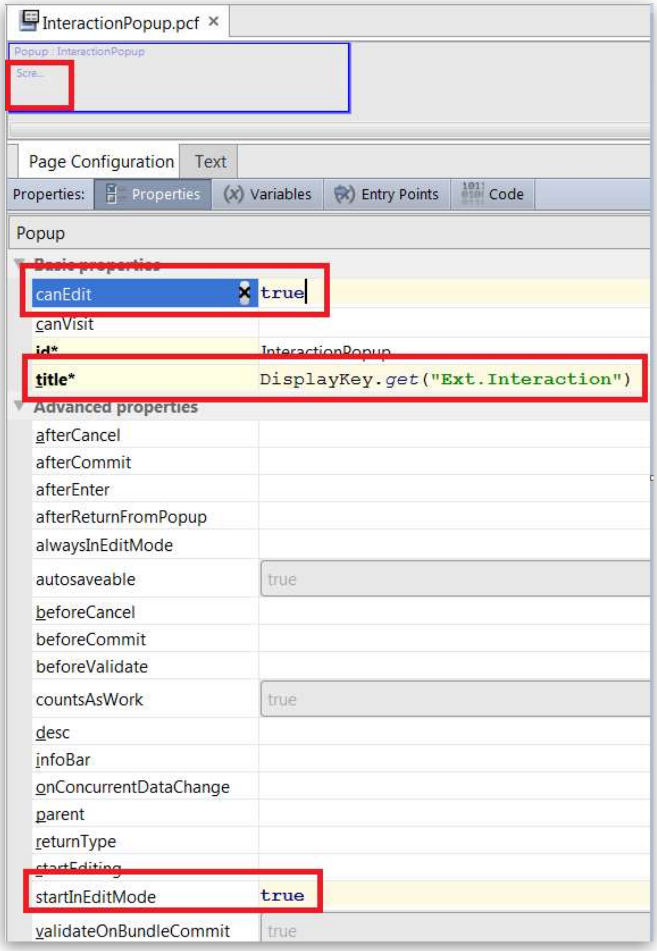


File name: Interaction Popup

Mode:

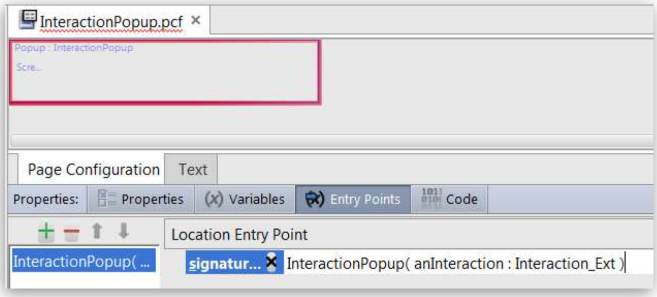
File type:
- Page
- Popup
- Accelerated Menu Actions
- Card View
- Chart View
- Detail View

3. **Add an inline Screen container and set the following properties: `startInEditMode`, `canEdit`, `title`**



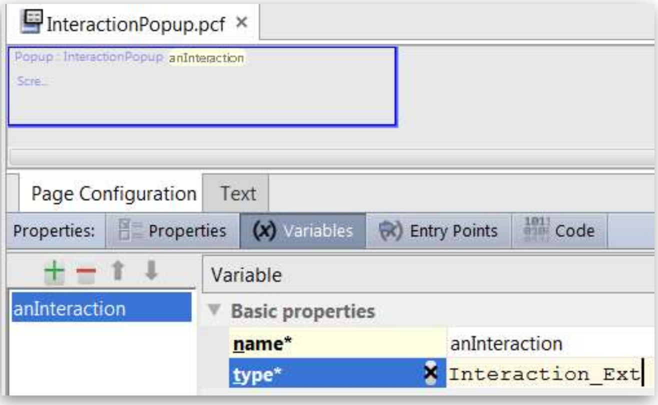
canEdit	true
canVisit	
id*	InteractionPopup
title*	DisplayKey.get("Ext.Interaction")
afterCancel	
afterCommit	
afterEnter	
afterReturnFromPopup	
alwaysInEditMode	
autosaveable	true
beforeCancel	
beforeCommit	
beforeValidate	
countsAsWork	true
desc	
infoBar	
onConcurrentDataChange	
parent	
returnType	
startEditing	
startInEditMode	true
validateOnBundleCommit	true

4. **Specify an entry point with a single Interaction_Ext parameter**



signatur... InteractionPopup(anInteraction : Interaction_Ext)

5. **Specify a variable for the parameter (same name and same type)**



name* anInteraction

type* Interaction_Ext

6. **Add a toolbar and edit buttons**

   a) From the toolbox, drag and drop a **Toolbar** into the screen you just added.

   b) Drag and drop an **Edit Buttons** widget into the toolbar.

7. **Add a detail view**

   a) Drag and drop a **Detail View** below the **Toolbar** widget

   b) The widget is colored red because the Detail View must contain an Input Column or a Table Layout. You'll fix this in the next step.

8. **Add an Input Column**

   a) Drag and drop an **Input Column** widget into the **Detail View**.

   b) Notice that the Detail View is no longer red.

9. **Add an input for the Interaction Date field**

   a) Drag and drop a **Date Input** into the **Input Column**.

   b) Set the editable property to **true**.

   c) Set the id property to **InteractionDate**.

   d) Set the label property to **DisplayKey.get("Ext.InteractionDate")**. This display key was created an earlier lab lab. If it does not exist, create it at this point.

   e) Set the value property to **anInteraction.InteractionDate**.

10. **Add an input for the Initiated by Contact field**

   a) Drag and drop a **Boolean Radio Button** Input into the **Input Column** below the Interaction Date input.

   b) Set the editable property to **true**.

   c) Set the id property to **InitiatedByContact**.

   d) Set the label property to **DisplayKey.get("Ext. InitiatedByContact ")**. This is a new display key, so create it using Studio.

   e) Set the value property to **anInteraction.InitiatedByContact**.

11. **Add an input for the Channel Type field**

   a) Drag and drop a **TypeKey Input** into the **Input Column** below the Initiated By Contact input.

   b) Set the editable property to **true**.

   c) Set the id property to **ChannelType**.

   d) Set the label property to **DisplayKey.get("Ext. ChannelType")**.

   e) Set the value property to **anInteraction.Channel**.

   f) Notice that the valueRange and valueType fields were automatically entered.

12. **Add an input for the Summary field**

   a) Drag and drop a **Text Input** into the **Input Column** below the Channel Type input.

   b) Set the editable property to **true**.

   c) Set the id property to **Summary**.

   d) Set the label property to **DisplayKey.get("Ext. Summary")**.

   e) Set the value property to **anInteraction.Summary**.

13. **Add an input for the Associated User field**

   a) Drag and drop an **Alt User Input** into the **Input Column** below the Summary input.

   b) Set the editable property to **true**.

   c) Set the id property to **AssociatedUser**.

d) Set the label property to **DisplayKey.get("Ext.AssociatedUser")**.

e) Set the value property to **anInteraction.AssociatedUser**.

14. Add an input for the Interaction Reason field

a) Drag and drop a **Range Input** into the **Input Column** below the Associated User input.

b) Set the editable property to **true**.

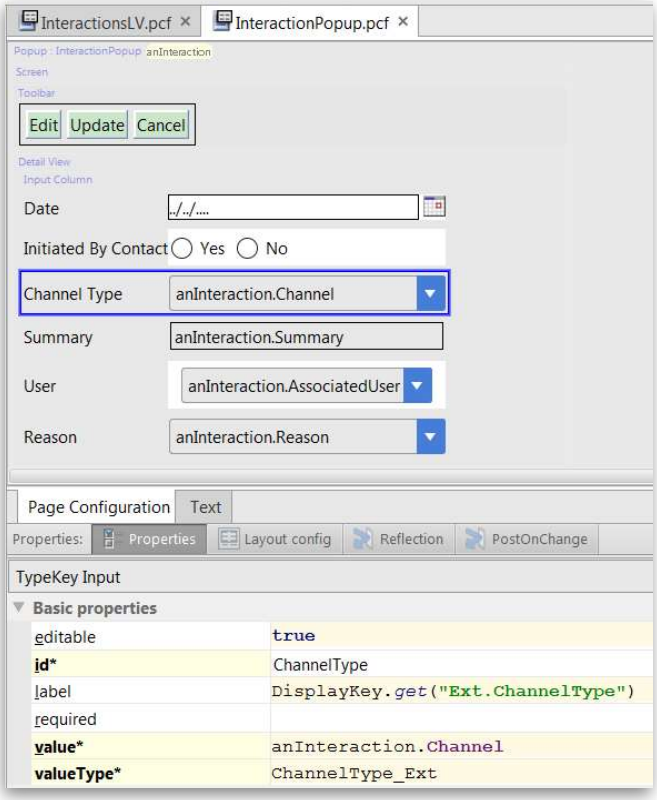
c) Set the id property to **ReasonType**.

d) Set the label property to DisplayKey.get("Ext.ReasonType").

e) Set the value property to **anInteraction.Reason**.

f) Notice that the valueRange and valueType fields were automatically entered.

15. The completed popup should look like this screenshot:



Date ../../....

Initiated By Contact ○ Yes ○ No

Channel Type anInteraction.Channel

Summary anInteraction.Summary

User anInteraction.AssociatedUser

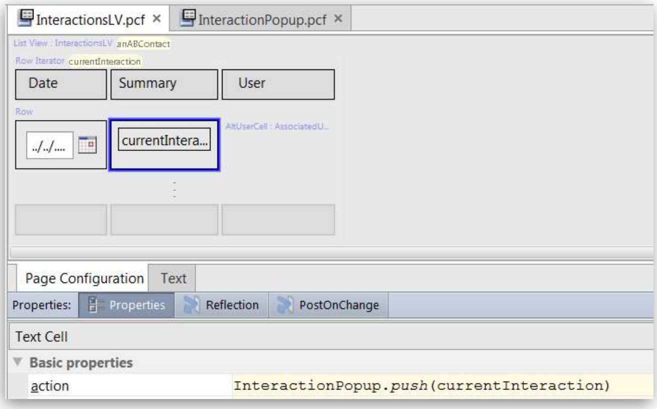
Reason anInteraction.Reason

TypeKey Input

Basic properties

editable	true
id*	ChannelType
label	DisplayKey.get("Ext.ChannelType")
required	
value*	anInteraction.Channel
valueType*	ChannelType_Ext

16. **Configure the action property of the Summary cell to open the InteractionPopup**



action `InteractionPopup.push(currentInteraction)`