Collect Contact Info from Your User

**Learning Objectives**

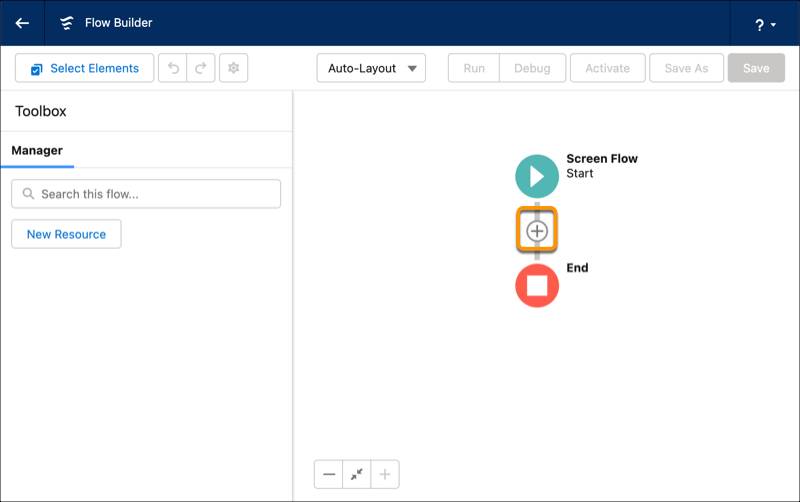
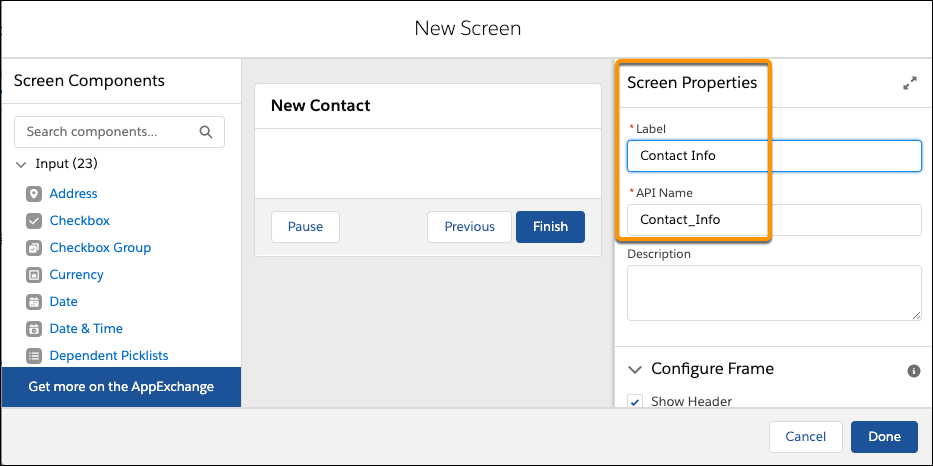
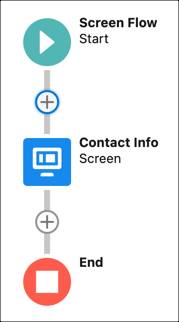
In this project, you’ll use Flow Builder to build a working flow that:

* Captures information from a user.
* Queries your Salesforce org for a record that matches what the user enters.
* Decides whether to update the matching record or create a new one.
* Creates or updates a record.

**Introduction**

This project walks you through the motions of building a simple flow. As you complete this project, don’t worry about understanding everything. Follow the instructions and observe how the various controls work in Flow Builder. By the end of this project, you’ll have a working flow. Later, you add to this flow as you complete the [Flow Builder](https://trailhead.salesforce.com/content/learn/modules/flow-builder) module. As you complete that module, you learn all about the flow building blocks that you use in this project. For now, gain some muscle memory as you repeat the flow-building pattern of drag or click, configure, and save. Let’s get started!

**Create the Flow and Add a Screen**

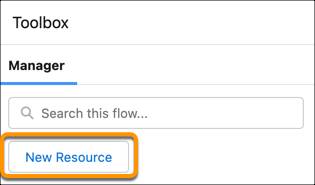
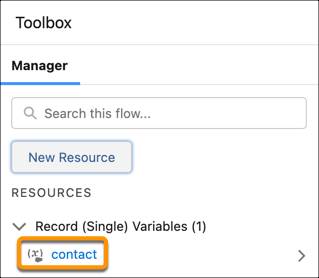
1. From Setup, enter Flows in the Quick Find box, then select **Flows**.
2. Click **New Flow**, select **Screen Flow**, and click **Create**.
3. From the canvas, click Add Element.  
   
4. Click **Screen**. The New Screen window opens.
5. Under Screen Properties for Label, enter Contact Info.
6. The API Name is automatically set to Contact\_Info. 
7. Click **Done**. The canvas now includes three items: Start, Contact Info, and End.  
   
8. Click **Save**, and enter these values.

| **Field** | **Value** |
| --- | --- |
| Flow Label | New Contact |
| Flow API Name | New\_Contact |

1. Click **Save**.

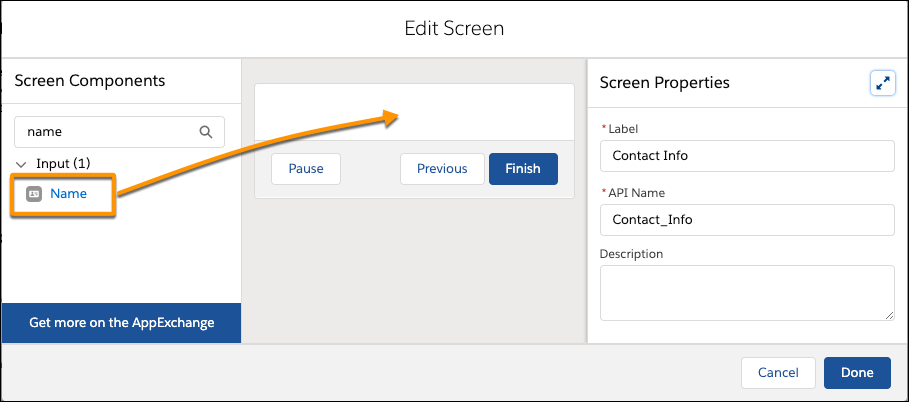
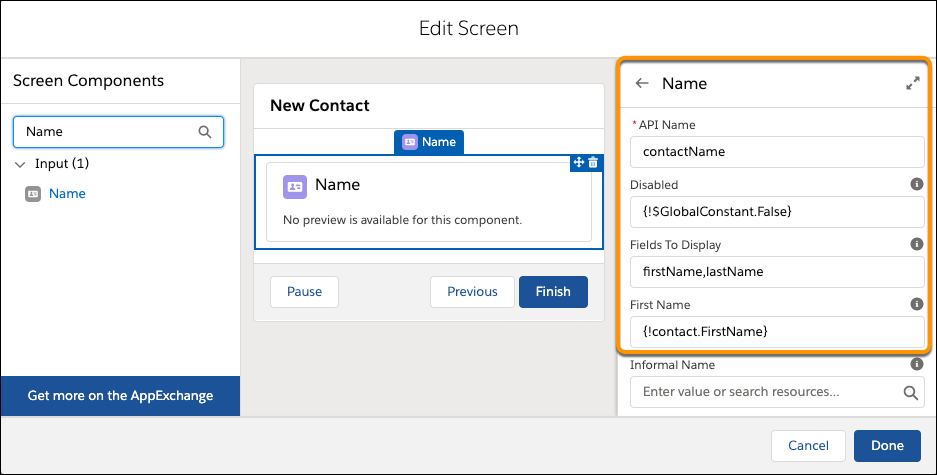
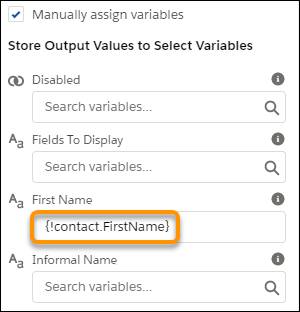
**Create a Record Variable for the Contact**

Because you plan to capture contact information from the user, let’s create a record variable to store that contact record data in the flow.

1. In the Manager tab, click **New Resource**.  
   
2. For Resource Type, select **Variable**.
3. For API Name, enter contact.
4. For Data Type, select **Record**.
5. For Object, enter **Contact**.
6. Click **Done**. In the toolbox, the Manager tab now contains a contact record variable resource.  
   
7. Click **Save**.

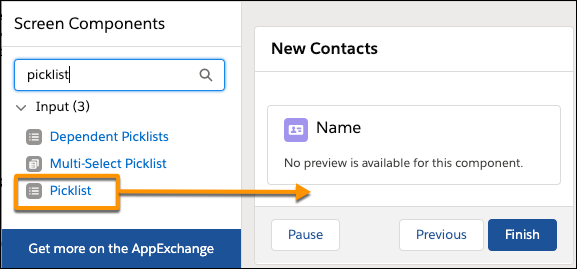
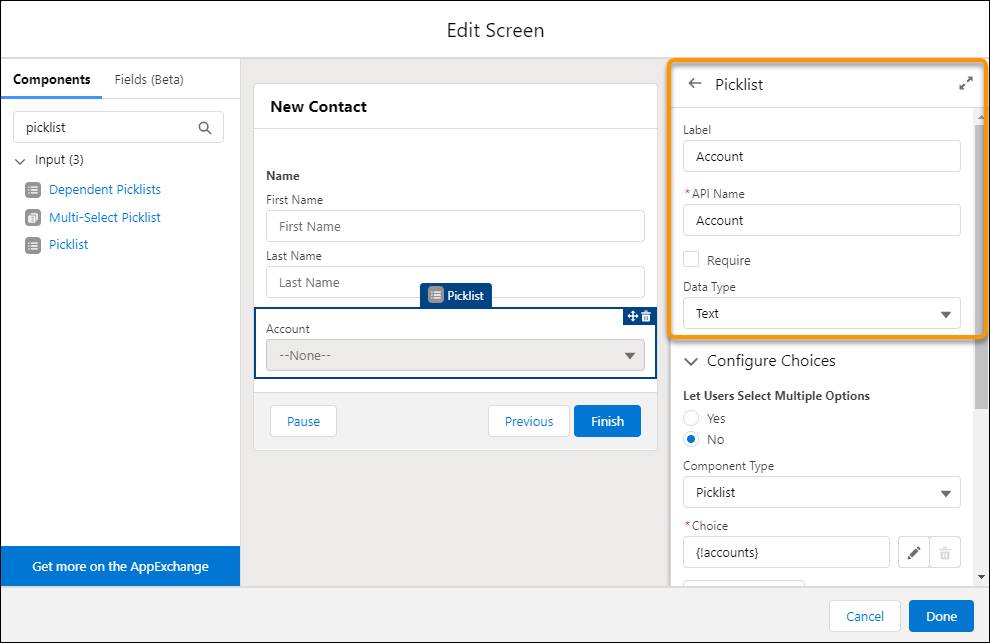
**Add a Name Input Component to the Screen**

You need to create form fields for the user to enter contact information on the screen. Let’s start with an input component that collects the contact’s name.

1. On the canvas, double-click the **Contact Info** screen. The Edit Screen window opens.
2. In the Components pane, enter Name in the search box.
3. Drag **Name** onto the screen canvas.  
     
   The properties pane now represents the Name input component.
4. In the properties pane, for API Name, enter contactName.
5. Scroll down and click **Advanced**. Make sure **Manually assign variables** is selected.
6. Complete these steps to take what the user enters in the First Name and Last Name fields and store those values in the contact record variable.
   1. For First Name, enter {!contact.FirstName}.  
      
   2. For Last Name, enter {!contact.LastName}.
7. Click **Done**.
8. Click **Save**.

**Add a Picklist Input Component to the Screen**

Now let’s add the picklist for the user to select the contact’s parent account.

1. On the canvas, double-click the **Contact Info** screen.
2. In the Screen Components pane, enter Picklist in the search box.
3. Drag Picklist onto the screen canvas just below the Name component.  
     
   The properties pane now represents the Picklist input component. 
4. In the properties pane, enter these values.

| **Field** | **Value** |
| --- | --- |
| Label | Account |
| API Name | Account |
| Data Type | Text |

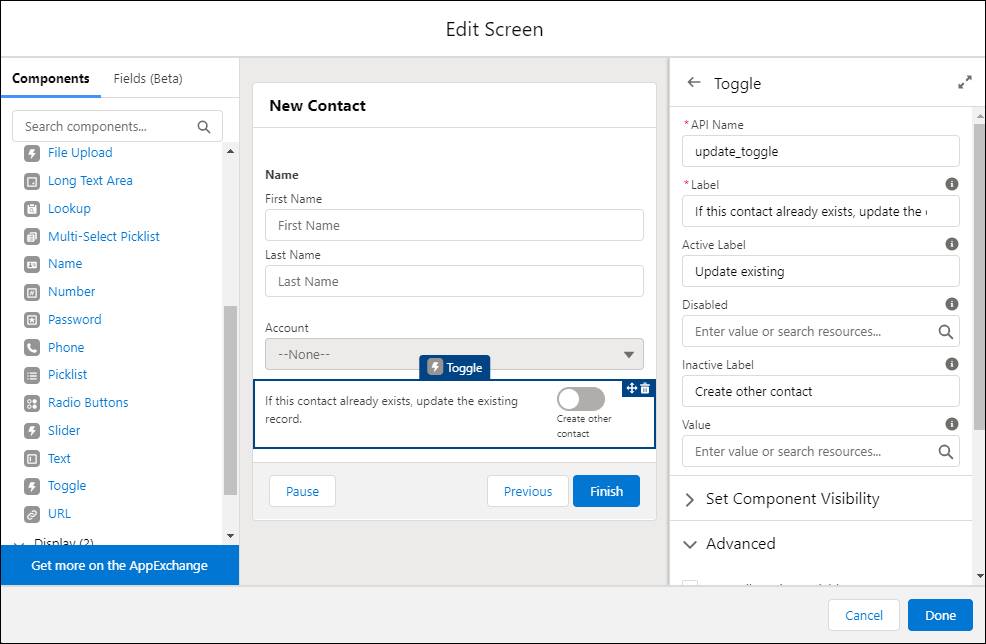
1. In the Configure Choices section, for Choice, select **New Choice Resource**. Enter these values.

| **Field** | **Value** |
| --- | --- |
| Resource Type | Record Choice Set |
| API Name | accounts |
| Object | Account |
| Condition Requirements | None—Get All Account Records |
| Choice Label | Name |
| Data Type | Text |
| Choice Value | Id |

1. In the Store More Account Field Values section, complete these steps to store the ID of the user-selected account in the contact record variable.
   1. For Field, select **Id**.
   2. For Variable, enter {!contact.AccountId}.
2. Click **Done** twice.
3. Click **Save**.

**Add a Toggle Input Component to the Screen**

Now let’s add the toggle for the user to specify whether (or not) to update an existing contact if one is found.

1. On the canvas, double-click the **Contact Info** screen.
2. From the Screen Components pane, drag Toggle onto the screen canvas just below the Account picklist component. 
3. In the properties pane, enter these values.

| **Field** | **Value** |
| --- | --- |
| API Name | update\_toggle |
| Label | If this contact already exists, update the existing record. |
| Active Label | Update existing |
| Inactive Label | Create other contact |
| Manually assign variables (under Advanced) | unchecked |
| Revisited Screen Values (under Advanced) | Use values from when the user last visited this screen |

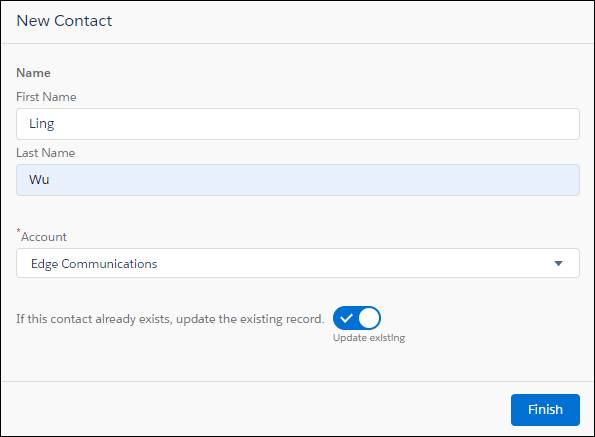
1. Click **Done**.
2. Click **Save**.

**Close and Open the Flow**

1. To exit Flow Builder, click the back arrow.  
   The flow list page appears.
2. To reopen the flow in Flow Builder, find New Contact on the flow list page, and click it.

**Run the Flow to View the Screen**

At this point, your flow can collect contact information from the user and store that information in flow variables. Let’s see what the screen looks like to users who run the flow.

1. Above the canvas, click **Run**.  
   
2. Enter any values, select any account from the list, and click **Finish**.  
   

Because the flow has no elements to execute after the screen, nothing happens when the flow finishes. The flow simply runs again, which means that you see the screen again.

Congrats on building and even running a flow! So far, the flow only displays a screen that accepts user input. Next, you configure the flow to compare some of that user input with data in your org.

Top of Form

**Verify Step**