Cloud Flow Designer Workbook

Cloud Flow Designer Workbook, Spring '16

@salesforcedocs

Last updated: April 27, 2016

CONTENTS

CLOUD FLOW DESIGNER WORKBOOK

New to flows? Work through these tutorials to get an introduction to building and distributing flows. Each tutorial stands on its own, so you can start with whichever one best suits your business needs.

Visual Workflow lets you build applications, called *flows*, to help users within your organization perform tasks easily and efficiently. For example, use Visual Workflow to script calls for a customer support center or to generate real-time quotes for a sales organization. This workbook offers a sampling of what Visual Workflow can do and—as you'll soon discover—how it can be a powerful, versatile tool for streamlining the processes that you manage. The workbook also demonstrates a few of the ways in which flows can be distributed to your end users.

IN THIS SECTION:

Before You Begin a Flow Tutorial

Get acquainted with the sample scenario, what you can expect from each tutorial, and how to set up your work environment for the tutorials.

Measure Customer Satisfaction

In response to requests from customers, Acme Wireless recently began offering a Preferred Repair service. With Preferred Repair, customers get faster repairs, expedited shipping, and access to a dedicated support team. After six months of rapid growth, management is eager to get feedback from customers about the program. They've decided to gather feedback by using a short customer satisfaction survey, which will be conducted over the phone by representatives from the Acme Wireless Customer Relations team.

Calculate Discounts on Opportunities

Sales associates cite Acme Wireless' aggressive discount program as a key factor in closing new deals. Unfortunately, getting permission to offer a discount is cumbersome: associates must email management, wait for a reply, and then pass this information back to the customer. The process is slow and is allowing Acme's more nimble competitors to land clients being courted by Acme. Streamlining this process is a top priority, and your team has been called in to assist.

Reassign Accounts

As one of the fastest growing companies in the segment, Acme Wireless' employees frequently move on to new, expanded roles within their teams. When this happens, the manager reassigns an employee's accounts to teammates based on the account's rating—hot, warm, or cold—and lets the teammates know that they've inherited accounts. To save managers' time, you've been asked to automate this process.

Before You Begin a Flow Tutorial

Get acquainted with the sample scenario, what you can expect from each tutorial, and how to set up your work environment for the tutorials.

IN THIS SECTION:

Flows and the Cloud Flow Designer

Briefly review some of the basic terms used to talk about flows in the Cloud Flow Designer.

Sample Scenario for Each Flow Tutorial

This workbook puts you in the shoes of an admin at a fictional company: Acme Wireless.

What You'll Learn in Each Flow Tutorial

If you've worked with flows before and want to skip to what interests you, or are just wondering what's ahead, see which elements and concepts each tutorial contains.

A Work Environment for Your Flow Tutorials

These tutorials are meant to show you around the world that is Visual Workflow, so your production organization isn't the right place to work through them. Instead, use a Developer Edition organization.

Flows and the Cloud Flow Designer

Briefly review some of the basic terms used to talk about flows in the Cloud Flow Designer.

Flows are built from *elements*. Elements represent actions, such as presenting information to, or collecting information from, flow users. Elements can also query, create, update, and delete information in Salesforce. When you connect elements together in the Flow Designer, you create a *flow*, or a series of actions that work together to accomplish one or more tasks. *Resources* include variables, constants, or formulas and are used to store or manipulate data within the flow.

As you work through the tutorials, we recommend keeping a copy of the *Visual Workflow Guide* handy—it's a great reference that you can use if you decide that you'd like a more in-depth description of a particular element or resource.

Sample Scenario for Each Flow Tutorial

This workbook puts you in the shoes of an admin at a fictional company: Acme Wireless.

Acme Wireless is a large company that repairs devices such as mobile phones, tablets, and gaming consoles. The company has a number of brick-and-mortar locations where customers can drop off devices for repair. In addition, these storefronts sell device accessories like tablet covers, mobile phone covers, power cords, rechargers, and so on. During the past year, Acme's executive leaders decided to shift resources away from its brick and mortar stores and instead direct them to building a robust online presence. The good news: the bet is paying off. Revenue is breaking records, and the company is growing rapidly. This growth is straining teams throughout Acme, however, and the company desperately needs to use its resources, like Salesforce, more effectively. That's where you come in.

In these tutorials, you'll build flows to:

- Help customer relations representatives measure customer satisfaction with Preferred Repair, a popular new service tier.
- Give sales associates a quick way to calculate discounts on opportunities.
- Allow managers to easily reassign account ownership when an employee moves on to a new role at the company.

What You'll Learn in Each Flow Tutorial

If you've worked with flows before and want to skip to what interests you, or are just wondering what's ahead, see which elements and concepts each tutorial contains.

Tutorial	Elements Used	Concepts Illustrated	Distribution Method
Measure Customer Satisfaction	Decision, Record Create, Screen	Conditional branching	Web tab
Calculate Discounts on Opportunities	Assignment, Decision, Record Lookup, Record Update	Conditional branching, autolaunched flows	Custom button
Reassign Accounts	Assignment, Decision, Fast Lookup, Fast Update, Loop, Screen, Send Email	Conditional branching, looping	Visualforce tab

A Work Environment for Your Flow Tutorials

These tutorials are meant to show you around the world that is Visual Workflow, so your production organization isn't the right place to work through them. Instead, use a Developer Edition organization.

If you don't have a Developer Edition organization, go to `http://sforce.co/1om1gHf` and sign up for a free Developer Edition account. Developer Edition is a fully functional version of Salesforce that you can use to develop Salesforce apps. Since it's free, there are limits on the amount of users, bandwidth, and storage you're allowed, but it includes all the features in Salesforce.

After that, you'll need to install the enhanced Warehouse data model into your organization.

IN THIS SECTION:

Install the Enhanced Warehouse Data Model

Some of the flow tutorials depend on certain objects and fields. Instead of manually adding that metadata to your organization, import the Warehouse data model.

You might be familiar with the Warehouse app if you've gone through the tutorials in the *Force.com Workbook*. We took the Warehouse data from that guide and added a few extra things to it to help show you what the Cloud Flow Designer can do.

1. Go to `www.salesforce.com` and log in to Salesforce using your Developer Edition credentials.
2. Open a new browser tab or window, and navigate to `https://github.com/forcedotcom/Salesforce1-Dev-Guide-Setup-Package`.

   Do this in the browser that you logged in to your developer organization with.
3. Open the README file.
4. Click the `bit.ly` link in the README file.

   This is the installation link for the enhanced Warehouse data package. You should be taken directly into your development organization to the Package Installation Details page.
5. Click **Continue**.
6. Click **Next**.
7. Click **Next**.
8. Click **Install**.
9. Wait for the installation to finish.
10. From the Force.com app menu, select Warehouse.
11. Click the Data tab.
12. Click **Create Data**.

That's it—you've now configured your Developer Edition organization and you're all set to start working through the tutorials. Let's get going!

Measure Customer Satisfaction

In response to requests from customers, Acme Wireless recently began offering a Preferred Repair service. With Preferred Repair, customers get faster repairs, expedited shipping, and access to a dedicated support team. After six months of rapid growth, management is eager to get feedback from customers about the program. They've decided to gather feedback by using a short customer satisfaction survey, which will be conducted over the phone by representatives from the Acme Wireless Customer Relations team.

In this tutorial, you'll build a flow that guides customer relations representatives through a survey script, allows them to log customer feedback, and generates a record of the call that management can review later. Then, you'll implement the flow by adding a custom tab that will appear at the top of the Salesforce app.

Before you dive into this tutorial, make sure that you have:

- Signed up for a Developer Edition organization at developer.salesforce.com/signup.
- Installed the enhanced Warehouse data model.

IN THIS SECTION:

Step 1: Ask the Customer to Participate in the Survey

The first thing the representative needs to do is ask the customer to participate in the satisfaction survey. To ensure that each customer has a consistent experience, management has prepared a script for the representative to read.

In this step, you'll configure a Screen element to display the script. A Screen element is how a flow displays information to users, provides choice selections, and captures information from users. In this step, you'll use a Screen element to display the part of the script where the representative asks the customer to participate in the survey. The customer can answer yes or no, so you'll also configure the element to capture the customer's response.

1. From Setup, enter `Flows` in the `Quick Find` box, then select **Flows**.
2. Click **New Flow**.
3. From the Palette tab in the left pane, click and drag a Screen element onto the canvas.
4. For `Name`, type *`Greet Customer`*.
   The `Unique Name` field is automatically populated based on this entry.
5. Prompt the representative to ask the customer to participate in the survey.
   a. From the Add a Field tab, double-click **Display Text**.

      As its name suggests, a display text field is how a Screen element displays text to users. By default, text is not formatted. However, you can click T to open Cloud Flow Designer's built-in text editor, which allows you to change font size, color, and perform other basic formatting.

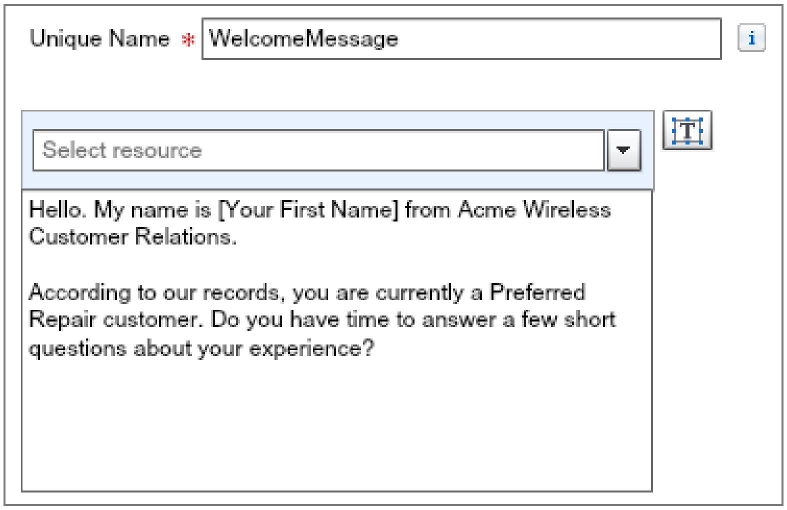
   b. Click **[Display Text]** in the preview pane.

   c. Configure the field.

Field	Value
`Unique Name`	*`WelcomeMessage`*

text box	*`Hello. My name is [Your First Name] from Acme Wireless Customer Relations.`*
	`According to our records, you are currently a Preferred Repair customer. We are very interested in hearing about your experience with this service. Do you have time to answer a few short questions?`

Here's what the field should look like.



Unique Name: WelcomeMessage

Hello. My name is [Your First Name] from Acme Wireless Customer Relations.

According to our records, you are currently a Preferred Repair customer. Do you have time to answer a few short questions about your experience?

6. Add a field where the representative can record whether the customer wants to participate in the survey.

   There are several ways to capture information from a user in the Cloud Flow Designer. You could add an input field like a text box, which means the representative would have to type in the customer's response—this gives the representative flexibility, but isn't very efficient. The better option is one of Cloud Flow Designer's choice fields, like radio buttons, which presents users with a list of selections to choose from.

   a. From the Add a Field tab, double-click **Radio Buttons**.

      Tip: Use radio buttons instead of a drop-down list whenever you present the user with a small number of choices. Here, the representative is limited to two customer responses: yes, the customer wants to participate in the survey, or no, they don't.

   b. Click **[Radio Buttons]** in the preview pane.

   c. For `Label`, type *`Customer Response`*.

      This is the description that the user will see next to the text box.

      The `Unique Name` field is automatically populated based on this entry.

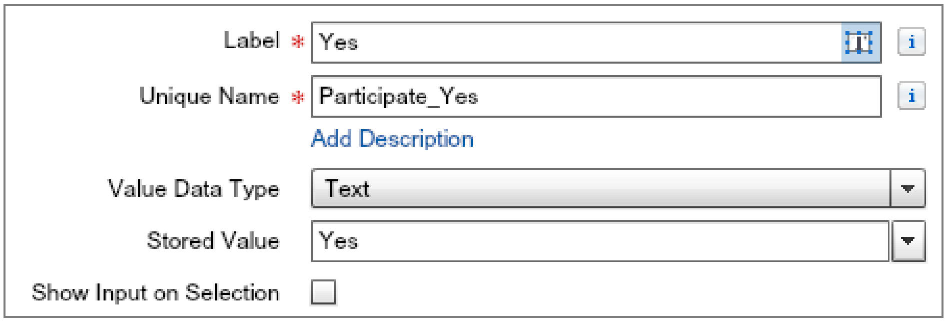
   d. For the `Choice` field, click ▼ and select **CREATE NEW** > **Choice**.

      With choice fields, you need to create the choice options that are available in the field. This is the choice option that the representative will select if the customer agrees to participate in the survey.

   e. Enter the following values.

Field	Value
`Label`	*`Yes`*
`Unique Name`	*`Participate_Yes`*
`Value Data Type`	*`Text`*
`Stored Value`	*`Yes`*

Here's what the field should look like.



Label *	Yes
Unique Name *	Participate_Yes
	Add Description
Value Data Type	Text
Stored Value	Yes
Show Input on Selection	□

f. Click **OK**.

g. Click **Add Choice**.

h. For the `Choice` field, click ▼ and select **CREATE NEW** > **Choice**.

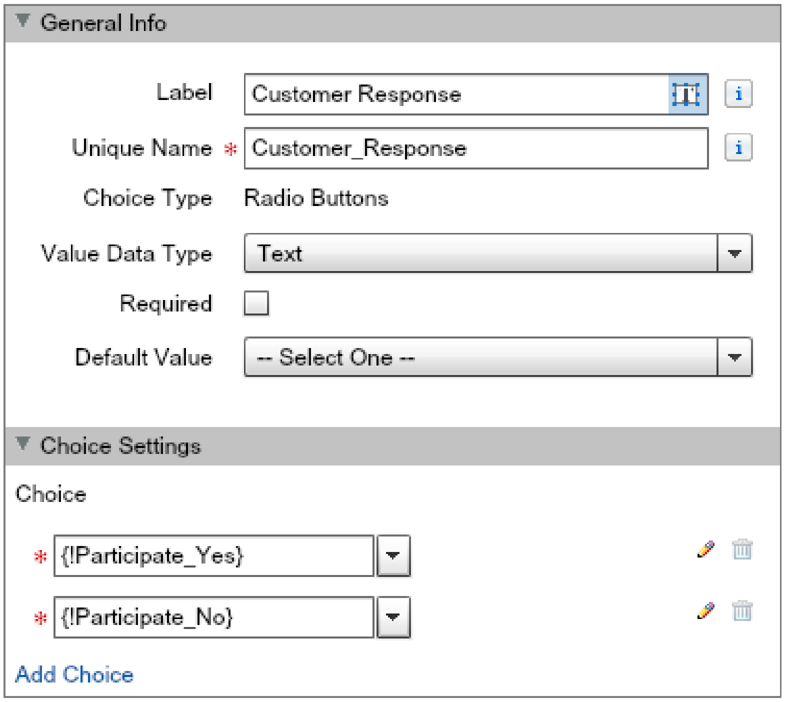
This is the choice the representative will select if the customer doesn't want to participate in the survey.

i. Enter the following values.

Field	Value
`Label`	*`No`*
`Unique Name`	*`Participate_No`*
`Value Data Type`	*`Text`*
`Stored Value`	*`No`*

j. Click **OK**.

7. Verify that your screen overlay looks like this, and then click **OK**.



▼ General Info

Label	Customer Response
Unique Name *	Customer_Response
Choice Type	Radio Buttons
Value Data Type	Text
Required	□
Default Value	-- Select One --

▼ Choice Settings

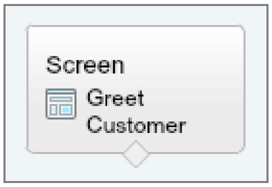
Choice

* {!Participate_Yes}

* {!Participate_No}

Add Choice

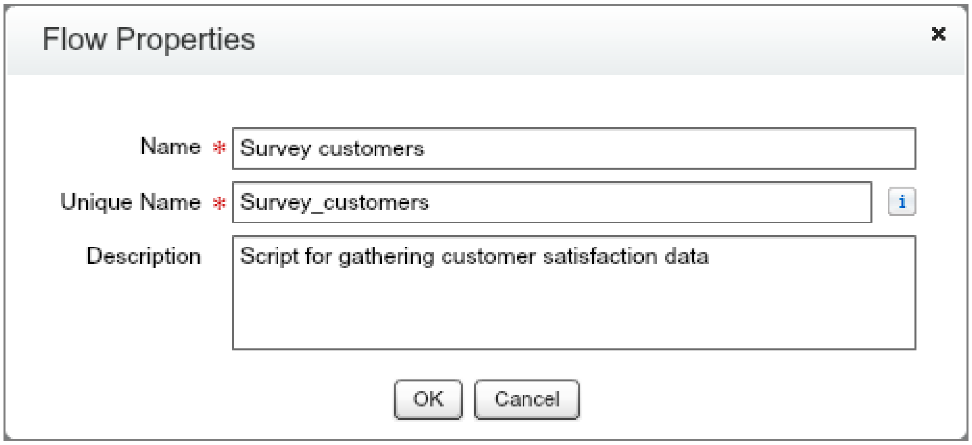
Your flow should now look something like this.



Step 2: Save the Flow

Once you've configured the first element in your flow, it's a good idea to save. Saving early and often safeguards your flow against unexpected events, such as losing your internet connection, and ensures that you won't have to repeat work that you've already done.

1. On the button bar, click **Save**.
2. For `Name`, type *`Survey customers`*.
   The `Unique Name` field is automatically populated based on this entry.
3. For `Description`, type *`Script for gathering customer satisfaction data`*.
4. Click **OK**.



Flow Properties

Name *	Survey customers
Unique Name *	Survey_customers
Description	Script for gathering customer satisfaction data

5. Ignore any warnings that appear, and click **OK**.

Step 3: Direct the Representative to the Right Place in the Script

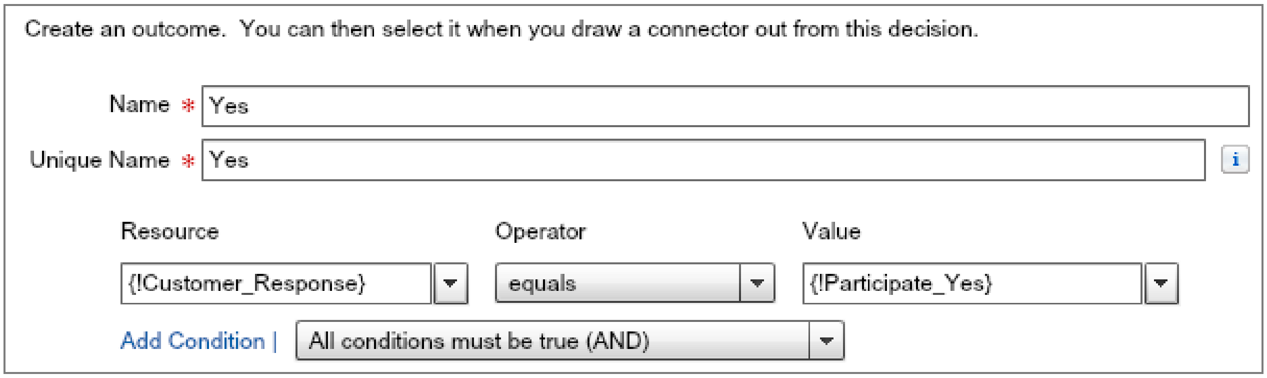
Next, the representative needs to be routed to the right place in the script based on whether the customer agreed to participate in the survey or not.

In this step, you'll configure a Decision element to steer the representative to a set of questions if the customer wants to participate in the survey, or a thank you script if they don't. A Decision element performs sorting and routing in a flow. Its job is analogous to a worker directing traffic on a road: using vehicle size as a guide, the worker might direct trucks and buses to the left lane, while routing cars and motorcycles to the right. Similarly, a Decision element routes a flow along one or more paths, known as *outcomes*, based on a set of criteria you determine. In this step, the Decision element will look at the value that's stored in the `Customer Response` choice field to decide on an outcome. In this tutorial, two outcomes are possible: customers either want to participate in the survey or they don't. If a customer decides to participate, the flow routes the representative to a Screen element that displays a list of survey questions. If a customer decides not to participate, the flow routes the representative to a different Screen element that presents a thank you script.

1. From the Palette tab, click and drag a Decision element onto the canvas.
2. Type *`Want to Participate?`* into the `Name` field.
   The `Unique Name` field is automatically populated based on this entry.
3. Create an outcome for customers who want to participate in the survey.
   a. Under the Outcomes section, for `Name`, type **Yes**.
      The `Unique Name` field is automatically populated based on this entry.
   b. Set the outcome criteria so that the flow takes this path only if the customer opted in to the survey.

Field	Value
`Resource`	*`{!Customer_Response}`*
`Operator`	*`equals`*
`Value`	*`{!Participate_Yes}`*

Here's what the outcome should look like.



Create an outcome. You can then select it when you draw a connector out from this decision.

Name * Yes

Unique Name * Yes

Resource	Operator	Value
{!Customer_Response}	equals	{!Participate_Yes}

4. Configure an outcome for customers who don't want to participate in the survey.

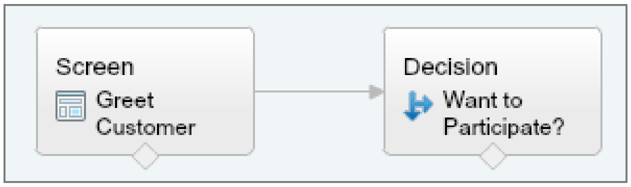
   a. Click **[Default Outcome]**.

      In a Decision element, the default outcome occurs when the criteria for the other outcomes aren't met. If the customer doesn't want to participate, the `Yes` outcome is false. So, the only alternative—*not* participating—becomes the default outcome.

   b. For `Name`, replace the predefined value with *`No`*.

5. Click **OK**.
6. Drag the node from the bottom of Greet Customer onto Want to Participate? to connect them.

Your flow should now look something like this.



Step 4: Survey the Customer

If the customer decides to participate in the survey, the flow directs the representative to a short set of survey questions.

In the following steps, you'll configure a Screen element to display questions related to the customer's experience with Preferred Repair. You'll also create fields where the representative can record the customer's responses.

IN THIS SECTION:

Display the Script with a Screen Element

In this step, you'll create a Screen element to display the survey questions and give the representative a place to confirm the name of their employer. You'll use the employer's name to generate the name for a survey record later in the flow.

Capture Service Quality Opinions with a Choice Field

Since one of the selling points of Preferred Repair is exceptional customer service, management wants to know if the Preferred Repair team is meeting this standard. So, the representative needs a place to log how the customer perceives service quality—you can do this using a choice field. Customers can rate their experience as excellent, needs improvement, or indicate that they have no opinion.

Capture Overall Satisfaction Level with a Choice Field

Next, the representative needs to ask about the customer's overall satisfaction with the service. For this, you'll need another choice field. Customers have five options to choose from, so to save screen space, you'll present the choices as a drop-down list.

Log Customer Comments with a Text Input Field

At the end of a satisfaction survey, it's a good idea to let customers share any opinions they couldn't mention elsewhere. So, in this step, the representative needs to ask the customer if they want to share any other comments. For this, you'll add another text input field.

Display the Script with a Screen Element

In this step, you'll create a Screen element to display the survey questions and give the representative a place to confirm the name of their employer. You'll use the employer's name to generate the name for a survey record later in the flow.

1. From the Palette tab, click and drag a Screen element onto the canvas.
2. For `Name`, type *`Survey Customer`*.
   The `Unique Name` field is automatically populated based on this entry.
3. Prompt the representative to thank the customer for participating in the survey.

   As before, you'll present this text to the representative by using a Display Text field.

   a. From the Add a Field tab, double-click **Display Text**.

   b. Click **[Display Text]** in the preview pane.

   c. Enter the following values.

Field	Value
`Unique Name`	*`ParticipationThanks`*
text box	*`Thank you for agreeing to participate in our survey.`*

4. Add an input field for the representative to enter the name of the company that the customer works for.

   Since the representative will enter the name of the company as a short string, a text box is the best choice here.

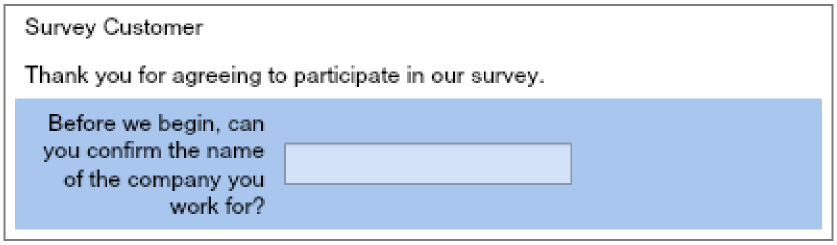
   a. From the Add a Field Tab, double-click **Textbox**.

   b. Click **[Textbox]** in the preview pane.

   c. Configure the field as follows.

Field	Value
`Label`	*`Before we begin, can you confirm the name of the company you work for?`*
`Unique Name`	*`Company_Name`*

5. Verify that the preview pane for the Screen element looks like this.



Survey Customer

Thank you for agreeing to participate in our survey.

Before we begin, can you confirm the name of the company you work for?

Capture Service Quality Opinions with a Choice Field

Since one of the selling points of Preferred Repair is exceptional customer service, management wants to know if the Preferred Repair team is meeting this standard. So, the representative needs a place to log how the customer perceives service quality—you can do this using a choice field. Customers can rate their experience as excellent, needs improvement, or indicate that they have no opinion.

1. Add a choice field to capture opinions about the quality of service provided by the Preferred Repair team.

   Since the number of choices is small, radio buttons are the best option here.

   a. From the Add a Field tab, double-click **Radio Buttons**.

   b. Click **[Radio Buttons]** in the preview pane.

   c. Enter the following values.

Field	Value
Label	*Of the following options, how would you rate your interactions with the Preferred Repair team?*
Unique Name	*Service*

   d. For the `Choice` field, click ▼ and select **CREATE NEW** > **Choice**.

   e. Configure a choice option for excellent service.

Field	Value
Label	*Excellent*
Unique Name	*Service_Excellent*
Value Data Type	*Text*
Stored Value	*Excellent*

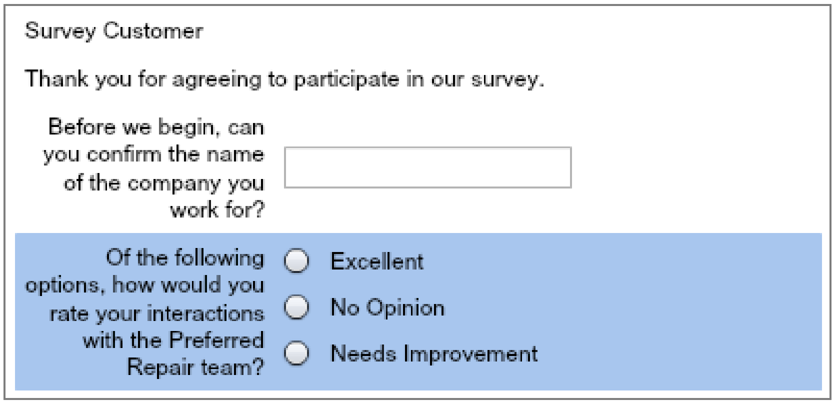
   f. Click **OK**.
   Now that you've got the hang of configuring a choice in detail, you'll repeat this process two more times.

   g. Click **Add Choice** twice to add choices for customers who have no opinion or feel that the service needs improvement.

   h. Configure the other choice options.

Choice	Label	Unique Name	Value Data Type	Stored Value
Choice 1	*No Opinion*	*Service_No_Opinion*	*Text*	*No Opinion*
Choice 2	*Needs Improvement*	*Service_Needs_Improvement*	*Text*	*Needs Improvement*

2. Verify that the preview pane for the Screen element looks like this.



Survey Customer

Thank you for agreeing to participate in our survey.

Before we begin, can you confirm the name of the company you work for?

Of the following options, how would you rate your interactions with the Preferred Repair team?

○ Excellent
○ No Opinion
○ Needs Improvement

Capture Overall Satisfaction Level with a Choice Field

Next, the representative needs to ask about the customer's overall satisfaction with the service. For this, you'll need another choice field. Customers have five options to choose from, so to save screen space, you'll present the choices as a drop-down list.

1. Add a choice field to capture customers' overall satisfaction level.

   a. From the Add a Field tab, double-click **Dropdown List**.

      Tip: Use a drop-down list instead of radio buttons whenever you present the user with a large number of options. Here, the customer has five options to choose from, so a drop-down list is the way to go.

   b. Click **[Dropdown List]** in the preview pane.

   c. Enter the following values.

Field	Value
Label	*Of the following options, how satisfied are you with Acme's Preferred Repair service?*
Unique Name	*Satisfaction*

2. Configure the choice options.

   a. For the `Choice` field, click ▼ and select **CREATE NEW** > **Choice**.

   b. Configure a choice option for when the customer is fully satisfied.

Field	Value
Label	*Satisfied*
Unique Name	*Satisfaction_Satisfied*
Value Data Type	*Text*
Stored Value	*Satisfied*

c. Click **OK**.
Now you'll repeat this process to add choice options for the other satisfaction levels.

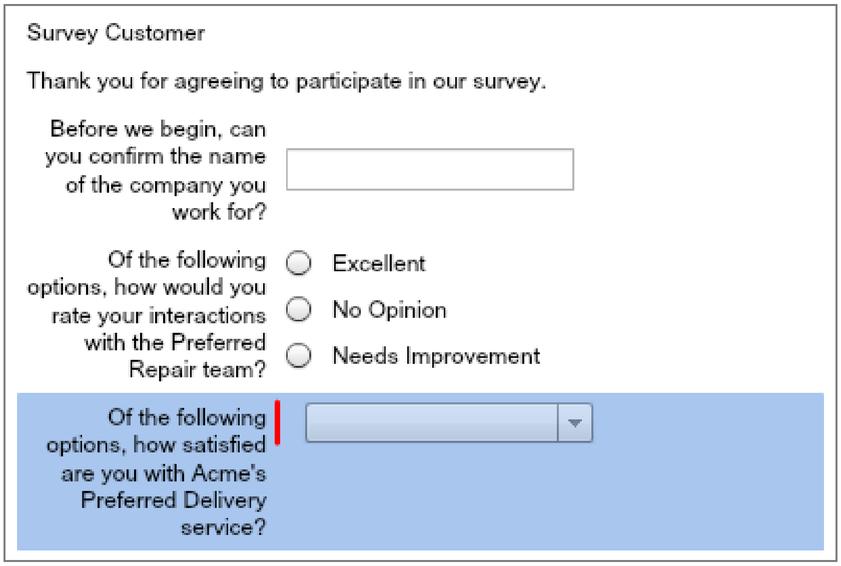
d. Click **Add Choice** four times.

e. Configure the choice options as follows.

Choice	Label	Unique Name	Value Data Type	Stored Value
Choice 1	*Somewhat Satisfied*	*Satisfaction_Somewhat_Satisfied*	*Text*	*Somewhat Satisfied*
Choice 2	*No Opinion*	*Satisfaction_No_Opinion*	*Text*	*No Opinion*
Choice 3	*Somewhat Unsatisfied*	*Satisfaction_Somewhat_Unsatisfied*	*Text*	*Somewhat Unsatisfied*
Choice 4	*Unsatisfied*	*Satisfaction_Unsatisfied*	*Text*	*Unsatisfied*

f. Click **OK**.

3. Verify that the preview pane looks like this.



Survey Customer

Thank you for agreeing to participate in our survey.

Before we begin, can you confirm the name of the company you work for?

Of the following options, how would you rate your interactions with the Preferred Repair team?
○ Excellent
○ No Opinion
○ Needs Improvement

Of the following options, how satisfied are you with Acme's Preferred Delivery service?

Log Customer Comments with a Text Input Field

At the end of a satisfaction survey, it's a good idea to let customers share any opinions they couldn't mention elsewhere. So, in this step, the representative needs to ask the customer if they want to share any other comments. For this, you'll add another text input field.

1. Add a text input field where representatives can type comments they receive from the customer.

   a. From the Add a Field tab, double-click **Long Text Area**.

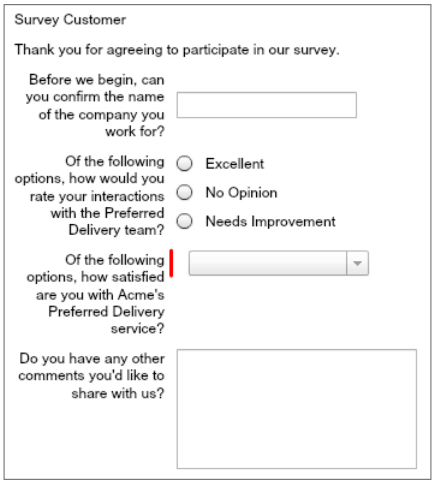
      Tip: Use a long text area if you expect your user will be typing more than one sentence into an input field. Because customer comments could be longer than one sentence, a long text area is the right choice here.

   b. Click **[Long Text Area]** in the preview pane.

   c. Enter the following values.

Field	Value
Label	*Do you have any other comments you'd like to share with us?*
Unique Name	*OtherComments*

2. Verify that the preview pane looks like this and then click **OK**



Survey Customer

Thank you for agreeing to participate in our survey.

Before we begin, can you confirm the name of the company you work for?

Of the following options, how would you rate your interactions with the Preferred Delivery team? ○ Excellent ○ No Opinion ○ Needs Improvement

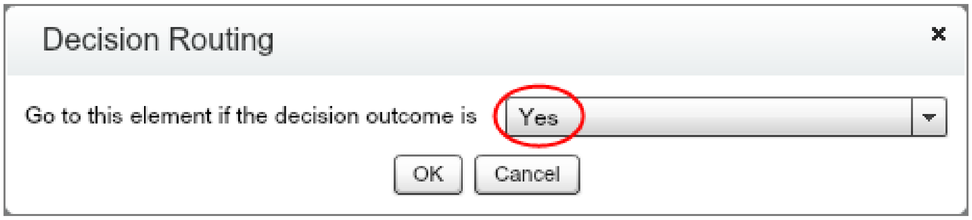
Of the following options, how satisfied are you with Acme's Preferred Delivery service?

Do you have any other comments you'd like to share with us?

3. Drag the node from the bottom of Want to Participate onto Survey Customer to connect them.

4. Verify that the Decision Routing overlay is set to **Yes**.

   This means that the representative will be directed to the survey questions only if the `Yes` outcome is true—that is, the customer agreed to participate in the first screen of the flow.

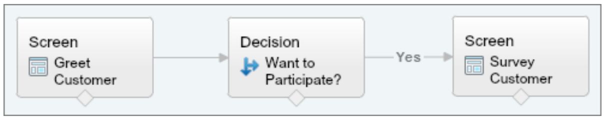


Decision Routing

Go to this element if the decision outcome is Yes

5. Click **OK**.
6. Save your progress.
   a. On the button bar, click **Save**.
   b. Ignore any warnings that appear, and click **OK**.

Your flow should now look something like this.



Step 5: Record the Survey Responses in Salesforce

Managers need to view the feedback in order to improve the service, so next you'll add the customer's responses to a record that managers can view in Salesforce.

To do this, you'll use a Record Create element. In a Record Create element, the flow creates a new record by using values that are stored in flow resources, like input fields. In this step, you'll use values from the flow to create a record for the custom Survey object that was included in the workbook support package.

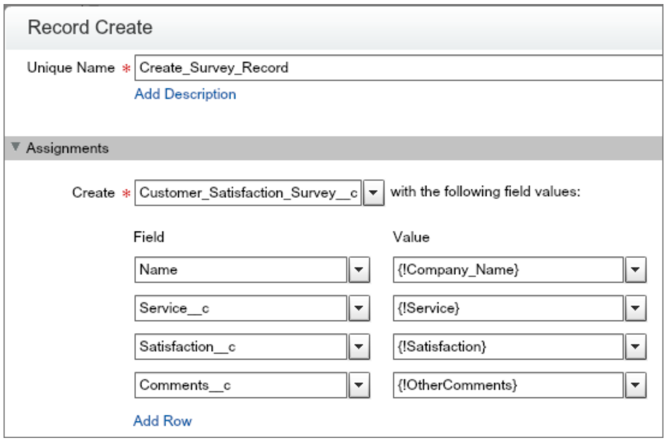
1. From the Palette tab, click and drag a Record Create element onto the canvas.

   Tip: There are two elements that allow you to create records in a flow: Record Create and Fast Create. A Fast Create element can only create a record from data stored in an sObject variable or sObject collection variable. In this flow, survey responses are determined by screen input and choice fields, so a Record Create element is the best option here.

2. For `Name`, type *`Create Survey Record`*.
   The `Unique Name` field is automatically populated based on this entry.
3. For `Create`, click and select **CUSTOM** > **Choice Customer_Satisfaction_Survey__c** from the picklist.
4. Assign the customer's responses to the survey record's fields.
   a. For `Field`, type *`Name`* and select **Name** from the picklist.
   `Name` tells managers which customer responded to the survey. You'll populate this field with the value stored in `Company_Name`, which is where the representative stored the name of the customer's employer earlier in the flow.
   b. For `Value`, type *`{!Company_Name}`* and select **{!Company_Name}** from the picklist.
   Now that you're comfortable assigning a value to a field, you'll repeat this process three more times.
   c. Click **Add Row** three times to set the values for the `Service`, `Satisfaction`, and `Comment` fields.
   d. Assign values to the fields.

Row	Field	Value
Row 2	*Service__c*	*{!Service}*
Row 3	*Satisfaction__c*	*{!Satisfaction}*
Row 4	*Comments__c*	*{!OtherComments}*

5. Verify that your overlay looks like this and then click **OK**.



Record Create

Unique Name * Create_Survey_Record

Add Description

▼ Assignments

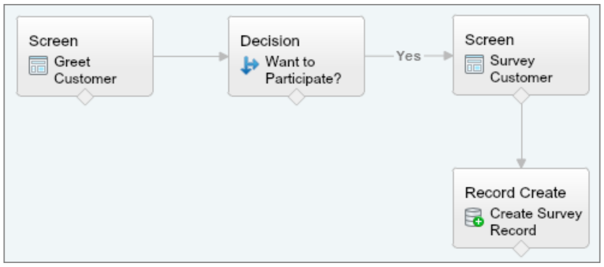
Create * Customer_Satisfaction_Survey__c with the following field values:

Field	Value
Name	{!Company_Name}
Service__c	{!Service}
Satisfaction__c	{!Satisfaction}
Comments__c	{!OtherComments}

Add Row

6. Drag the node from the bottom of Survey Customer onto Create Survey Record to connect them.

Your flow should now look something like this.



Step 6: Thank the Customer

Before ending the call, the representative thanks customers for their time and business. Just like in the first step of the flow, you'll present the script to the representative with a Screen element.

1. From the Palette tab, click and drag a Screen element onto the canvas.
2. For `Name`, type *`Thank Customer`*.
   The `Unique Name` field is automatically populated based on this entry.
3. Prompt the representative to thank the customer.
   a. From the Add a Field tab, double-click **Display Text**.

  b. Click **[Display Text]** in the preview pane.

  c. Configure the field.

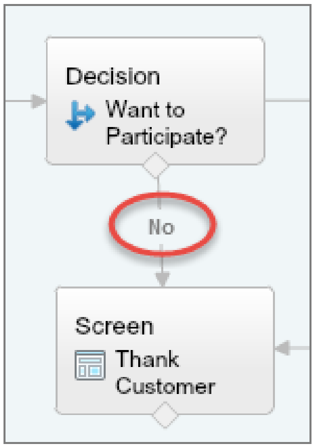
Field	Value
Unique Name	*ThankCustomer*
text box	*Thank you for your time. On behalf of everyone at Acme Wireless, we'd like to thank you for being a customer. Have a great day!*

  d. Click **OK**.

4. Finish routing the flow.

  a. Drag the node from the bottom of Create Survey Record onto Thank Customer to connect them.

  b. Drag the node from the bottom of Want to Participate? onto Thank Customer to connect them.

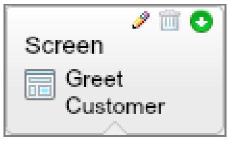


Notice that the word *No* now appears over the connector. Because you routed the `Yes` outcome in a previous step, the Cloud Flow Designer automatically assigns the only outcome left—the default outcome—to this path.

5. Designate a start element for the flow.

A start element tells a flow where to begin. If you try to save a flow without designating a start element, the Cloud Flow Designer displays an activation warning. While this won't keep you from saving, you must designate a start element before you activate or run your flow.

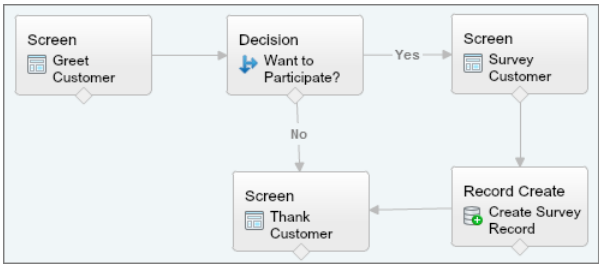
  a. Hover your mouse over Greet Customer, and three icons appear in the upper right corner.



  b. Click ⊕ to set Greet Customer as the start element.

6. On the button bar, click **Save**.

Your finished flow should look something like this.

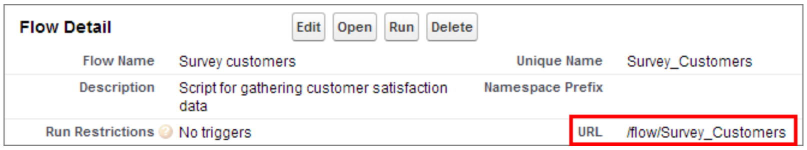


Step 7: Implement the Flow Using a Web Tab

You've built a flow that representatives can use to gather feedback about Preferred Repair, but they can't access it yet. So, implement the flow to make it easy for representatives to use when they're in Salesforce.

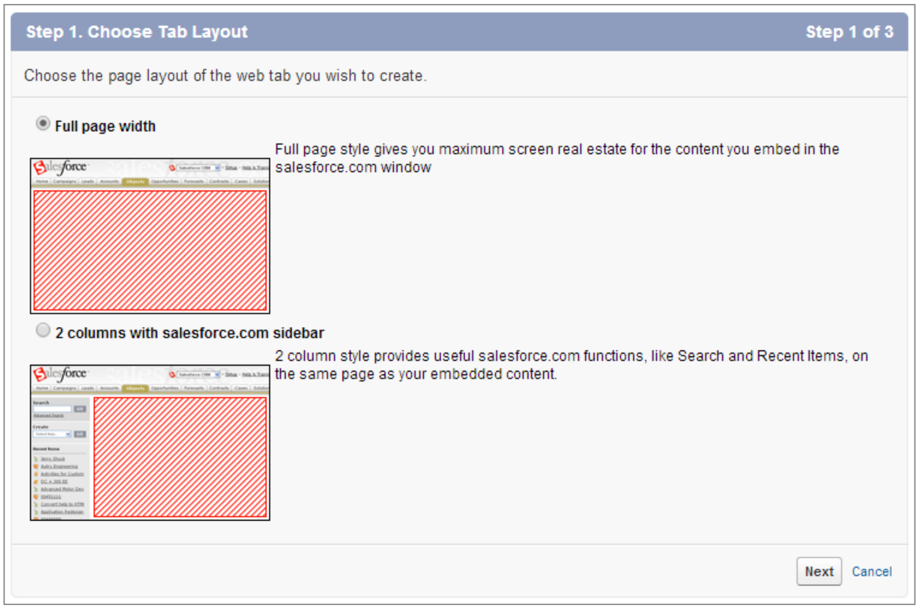
In this step, you'll implement this flow using a web tab, which is a web-based application that end users can access through a custom tab in the Salesforce app.

1. Get the URL of the flow.
   a. On the button bar, click **Close**.
   The Flow Designer closes and the Flow Detail page opens.
   b. Copy the flow's URL.



Flow Name	Survey customers	Unique Name	Survey_Customers
Description	Script for gathering customer satisfaction data	Namespace Prefix	
Run Restrictions	No triggers	URL	/flow/Survey_Customers

2. Create a web tab.
   a. From Setup, enter `Tabs` in the `Quick Find` box, then select **Tabs**.
   b. In the Web Tabs related list, click **New**.
   The New Web Tab wizard opens.
   c. Choose **Full Page Width**, which excludes the sidebar and gives users maximum screen space.
   Here's what the first page of the wizard should look like.



Step 1. Choose Tab Layout

Step 1 of 3

Choose the page layout of the web tab you wish to create.

● Full page width

Full page style gives you maximum screen real estate for the content you embed in the salesforce.com window

○ 2 columns with salesforce.com sidebar

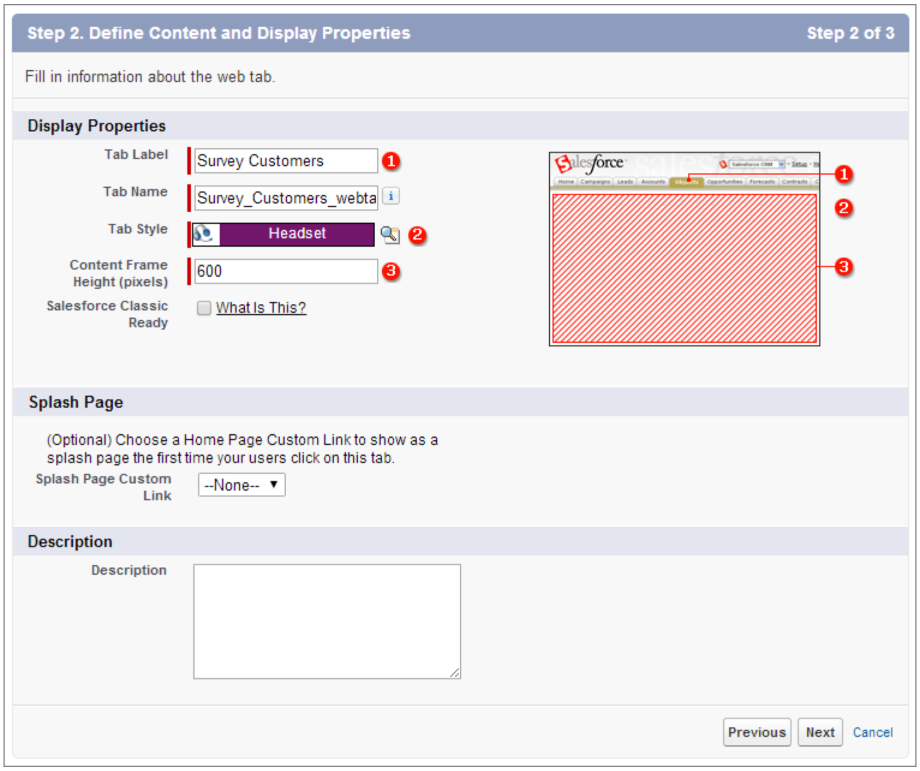
2 column style provides useful salesforce.com functions, like Search and Recent Items, on the same page as your embedded content.

d. Click **Next**.

e. Configure how the tab displays.

Field	Value
`Tab Label`	*`Survey Customers`* The `Tab Name` field is automatically populated based on this entry.
`Tab Name`	*`Survey_Customers_webtab`*
`Tab Style`	*`Headset`*
`Content Frame Height (pixels)`	*`600`*

Here's what the second page of the wizard should look like.



Step 2. Define Content and Display Properties — Step 2 of 3

Fill in information about the web tab.

Display Properties

Tab Label	Survey Customers (1)
Tab Name	Survey_Customers_webta
Tab Style	Headset (2)
Content Frame Height (pixels)	600 (3)
Salesforce Classic Ready	□ What Is This?

Splash Page

(Optional) Choose a Home Page Custom Link to show as a splash page the first time your users click on this tab.

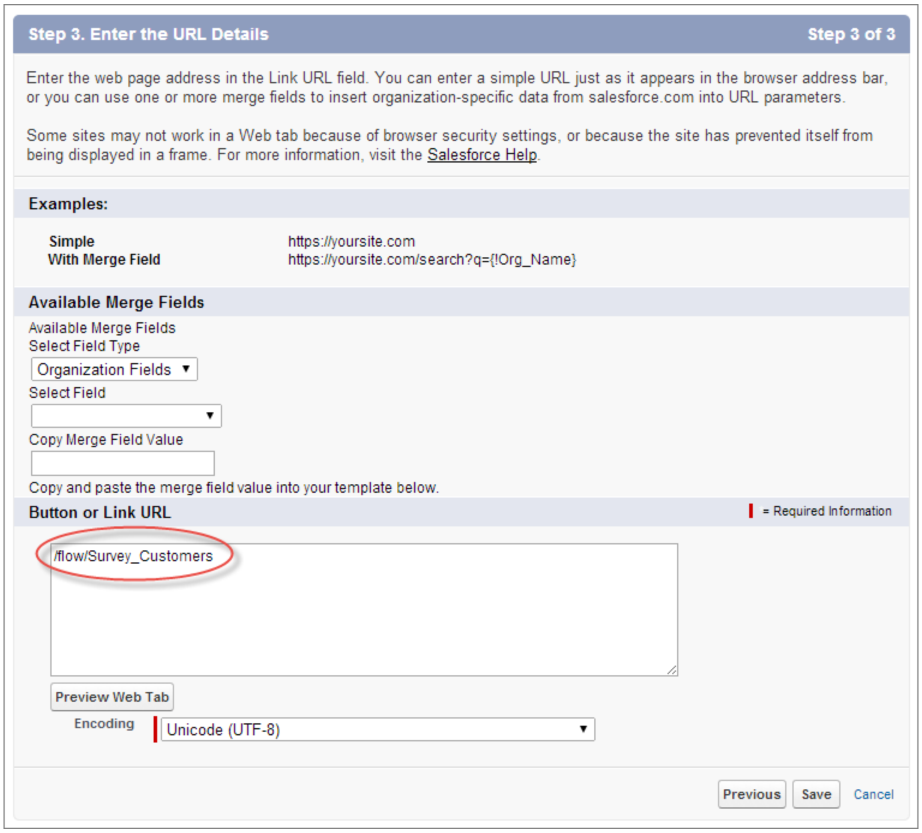
Splash Page Custom Link: --None--

Description

Description:

f. Click **Next**.

g. For `Button or Link URL`, delete the default content and paste the URL you copied earlier into the text box. Here's what the third page of the wizard should look like.



Step 3. Enter the URL Details Step 3 of 3

Enter the web page address in the Link URL field. You can enter a simple URL just as it appears in the browser address bar, or you can use one or more merge fields to insert organization-specific data from salesforce.com into URL parameters.

Some sites may not work in a Web tab because of browser security settings, or because the site has prevented itself from being displayed in a frame. For more information, visit the Salesforce Help.

Examples:

Simple	https://yoursite.com
With Merge Field	https://yoursite.com/search?q={!Org_Name}

Available Merge Fields

Available Merge Fields

Select Field Type: Organization Fields

Select Field

Copy Merge Field Value

Copy and paste the merge field value into your template below.

Button or Link URL = Required Information

/flow/Survey_Customers

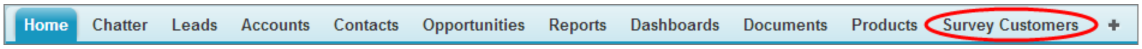
Encoding: Unicode (UTF-8)

h. Click **Next**.

i. Click **Next** again.

j. Click **Save**.

Survey Customers now appears next to the other tabs.



Step 8: Test the Flow

Testing is an important part of building a flow, allowing you to fine-tune the flow’s behavior, identify and fix bugs, and otherwise make sure your users have a pleasant experience. And of course, you benefit, since you’re much less likely to spend time later responding to support requests and fixing errors.

In the following steps, you’ll assume the role of a customer relations representative and create a test survey record. Then, you'll verify that the record was created successfully. Let’s get testing!

IN THIS SECTION:

Create a Fictional Survey Record

In this step, you'll walk through the flow just like a customer relations representative would. In so doing, you'll create a test survey record for Barbary Coast Wireless, one of Acme Wireless' largest customers.

Verify the Survey Record

In this step, you'll verify that the flow successfully created the survey record. Also, since Salesforce doesn't display custom objects by default, you'll configure a tab to display surveys.

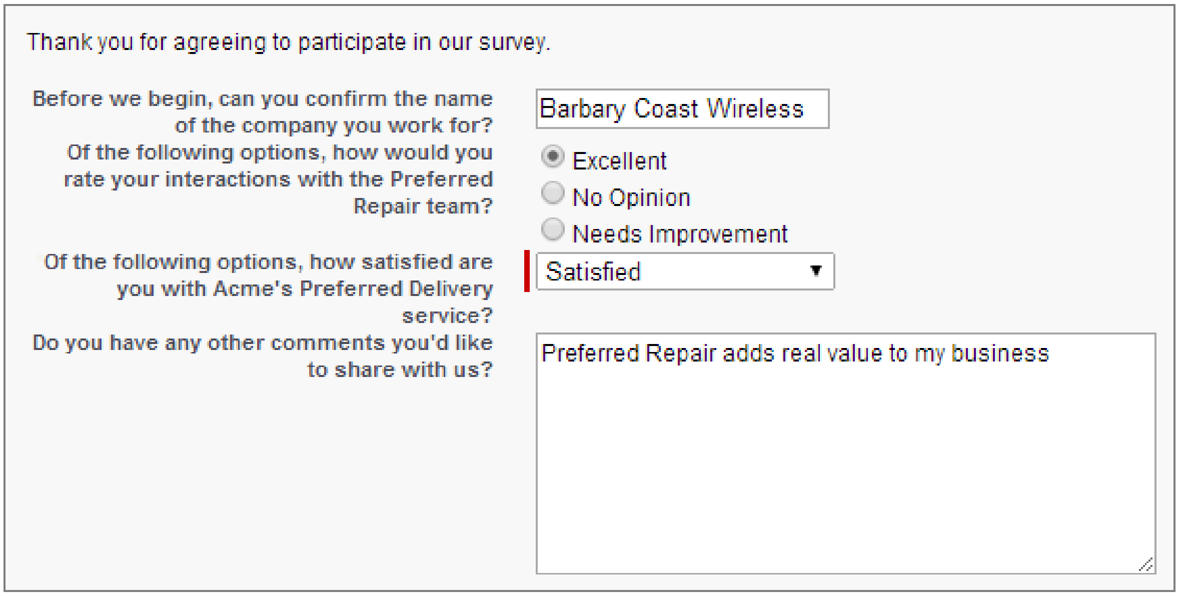
Create a Fictional Survey Record

In this step, you'll walk through the flow just like a customer relations representative would. In so doing, you'll create a test survey record for Barbary Coast Wireless, one of Acme Wireless' largest customers.

1. Click the Survey Customers tab.
2. Since you want to test that a record is created, for `Customer Response`, select **Yes**.
3. Respond to the survey as follows.

Question	Response
`Before we begin, can you confirm the name of the company you work for?`	*`Barbary Coast Wireless`*
`Of the following options, how would you rate your interactions with the Preferred Repair team?`	*`Excellent`*
`Of the following options, how would you rate your interactions with the Preferred Repair team?`	*`Satisfied`*
`Do you have any other comments you'd like to share with us?`	*`Preferred Repair adds real value to my business`*

Here's what the survey responses should look like.



Thank you for agreeing to participate in our survey.

Before we begin, can you confirm the name of the company you work for? Barbary Coast Wireless

Of the following options, how would you rate your interactions with the Preferred Repair team?
● Excellent
○ No Opinion
○ Needs Improvement

Of the following options, how satisfied are you with Acme's Preferred Delivery service? Satisfied ▼

Do you have any other comments you'd like to share with us? Preferred Repair adds real value to my business

4. Click **Next**.
5. Click **Finish**.

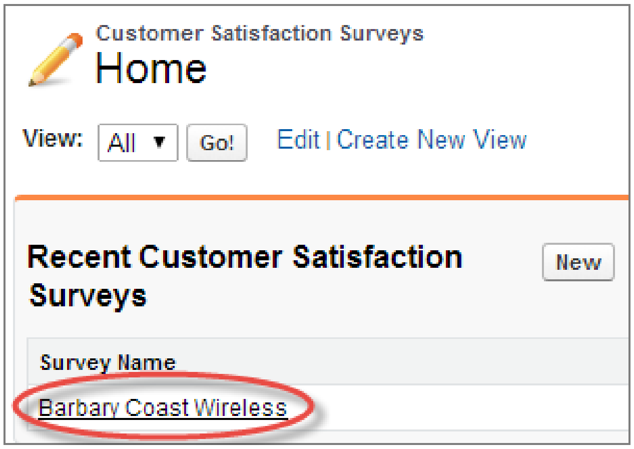
Verify the Survey Record

In this step, you'll verify that the flow successfully created the survey record. Also, since Salesforce doesn't display custom objects by default, you'll configure a tab to display surveys.

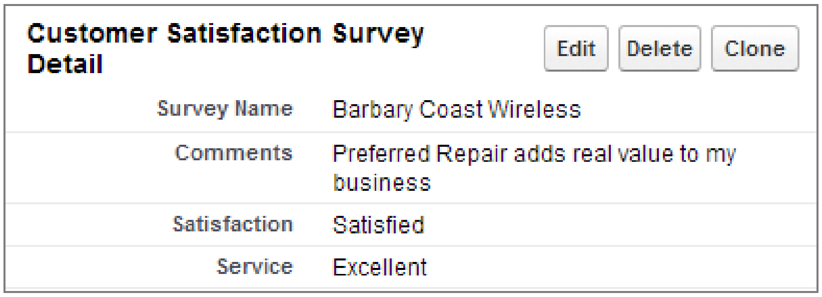
1. Add a tab to display surveys.
   a. From Setup, enter `Tabs` in the `Quick Find` box, then select **Tabs**.
   b. In the Custom Object Related list, click **New**.
   The Custom Tab wizard opens.
   c. Configure how the tab displays.

Field	Value
`Object`	*`Customer Satisfaction Surveys`*
`Tab Style`	*`Pencil`*
`Description`	*`Tab to display survey object records`*

   d. Click **Next**.
   e. Click **Next** again.
   f. Click **Save**.
2. Navigate to the survey record for Mike's Tablet Warehouse.
   a. Click the Customer Satisfaction Surveys tab.
   b. In the Recent Customer Satisfaction Surveys related list, click **Barbary Coast Wireless**.



3. Verify that your responses appear in the survey record.


Customer Satisfaction Survey Detail

Survey Name	Barbary Coast Wireless
Comments	Preferred Repair adds real value to my business
Satisfaction	Satisfied
Service	Excellent

Congratulations! You've just built, tested, and implemented your first flow.

If you haven't worked with flows before, you'll soon discover that existing flows make great starting points for new ones—particularly as your organization's needs grow and change. Take the customer satisfaction survey you just created, for example. If Acme Wireless wanted to quickly address cases where customers reported dissatisfaction with the service, you could configure the flow to email a customer relations manager on the Preferred Repair team. Or, to give the flow more power, you could associate the survey records you generate with the accounts in the Salesforce database. If you haven't already, check out the other tutorials in this workbook to learn even more about the Cloud Flow Designer!

Calculate Discounts on Opportunities

Sales associates cite Acme Wireless' aggressive discount program as a key factor in closing new deals. Unfortunately, getting permission to offer a discount is cumbersome: associates must email management, wait for a reply, and then pass this information back to the customer. The process is slow and is allowing Acme's more nimble competitors to land clients being courted by Acme. Streamlining this process is a top priority, and your team has been called in to assist.

In this tutorial, you'll create a flow that automatically determines how much of a discount to offer a customer. Unlike the previous flow, which customer service representatives interacted with directly as they navigated between survey screens, this one will work in the background, hidden from users. Flows that don't require user interaction are called *autolaunched flows*. This flow will be launched via a button on all opportunity records in Salesforce. When a sales associate clicks the button, information from the opportunity is passed into the flow, which then determines the right discount to offer. After, the flow updates the opportunity with the right discount percentage.

Before you dive into this tutorial, make sure that you have:

- Signed up for a Developer Edition organization at developer.salesforce.com/signup.
- Installed the enhanced Warehouse data model.

IN THIS SECTION:

Step 4: Determine the Right Discount to Apply

Next, the flow needs to determine which discount to apply. In the new discounting program, Acme Wireless will award discounts based on the annual revenue: the higher the revenue, the larger the discount.

Step 5: Assign Numerical Values to Discounts

At this point, the flow knows the annual revenue of the account and can use that value to direct the flow to the right discount. But the discounts—full, partial, or no—are abstract. To be useful, each discount needs to be linked to a numerical value—10 percent, 20 percent, and so forth.

Step 6: Update the Opportunity with the Discount

Once the flow has determined which discount to apply, it needs to return that information back to the opportunity so that the sales associate knows how much of a discount to offer the customer.

Step 7: Implement the Flow with a Custom Button

You've successfully created the flow, but it's only accessible from Setup. Let's implement the flow so that your sales representatives can easily find and use it.

Step 8: Customize the Opportunity Page Layout

Soon, you'll want to test the flow—just to make sure that everything works. Before you can do that, add the `Discount Percentage` and `Discounted Amount` fields to the opportunity page layout.

Step 9: Create a Sample Opportunity

Just one more thing before you can test the flow. Let's create a sample opportunity so that the flow has something to update.

Step 10: Test the Flow

Testing is an important part of building a flow, allowing you to fine-tune the flow's behavior, identify and fix bugs, and otherwise make sure your users have a pleasant experience. And of course, you benefit, since you're much less likely to spend time later responding to support requests and fixing errors.

Step 1: Create the Flow and Add Variables

In the first step, you'll create the flow and configure it to receive an account ID and opportunity ID, which the flow will use in later steps to determine which discount to apply.

1. From Setup, enter *`Flows`* in the `Quick Find` box, then select **Flows**.
2. Click **New Flow**.
3. Create a variable to store the opportunity ID.

   a. From the Resources tab in the left pane, double-click **Variable**.

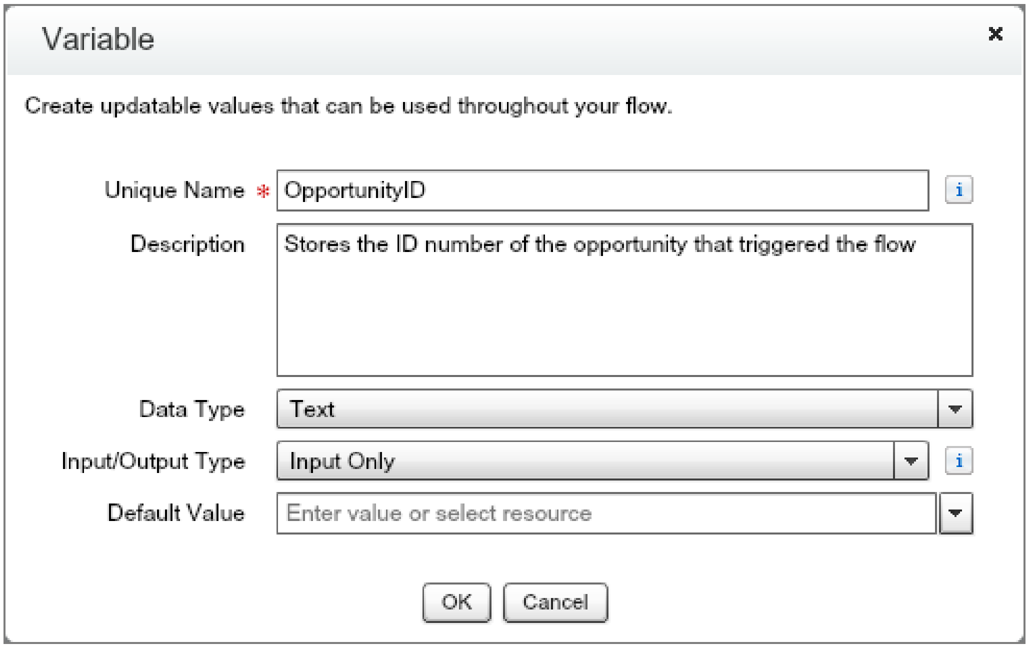
   > Note: You may remember that your sales representatives will use a custom button to launch the flow. When clicked, the custom button will call the flow using the flow URL. When you use this method, you can append extra information, like field values, to the end of the URL—these are known as *URL parameters*. URL parameters don't support sObject variables, however, so you need to use a variable.

   b. Configure the variable.

Field	Value
`Unique Name`	*`OpportunityID`*
`Description`	*`Stores the ID number of the opportunity that triggered the flow`*
`Data Type`	*`Text`*

Field	Value
`Input/Output Type`	*`Input Only`* Notice that the variable is set to `Input Only`. If it was set to `Private`, Salesforce wouldn't be able to pass data into it.

Here's what your overlay should look like.



Variable

Create updatable values that can be used throughout your flow.

Unique Name * OpportunityID

Description Stores the ID number of the opportunity that triggered the flow

Data Type Text

Input/Output Type Input Only

Default Value Enter value or select resource

c. Click **OK**.

4. Create a second variable to store the account ID.

   a. From the Resources tab in the left pane, double-click **Variable**.

      Just like before, this value will be passed into the flow as a URL parameter, so a variable is the right choice.

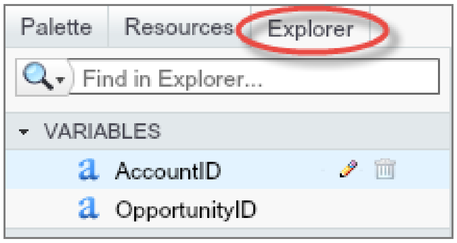
   b. Configure the variable.

Field	Value
`Unique Name`	*`AccountID`*
`Description`	*`Stores the ID number of the account associated with the opportunity`*
`Data Type`	*`Text`*
`Input/Output Type`	*`Input Only`*

c. Click **OK**.

5. Click the Explorer tab to verify that `OpportunityID` and `AccountID` have been added to the flow.

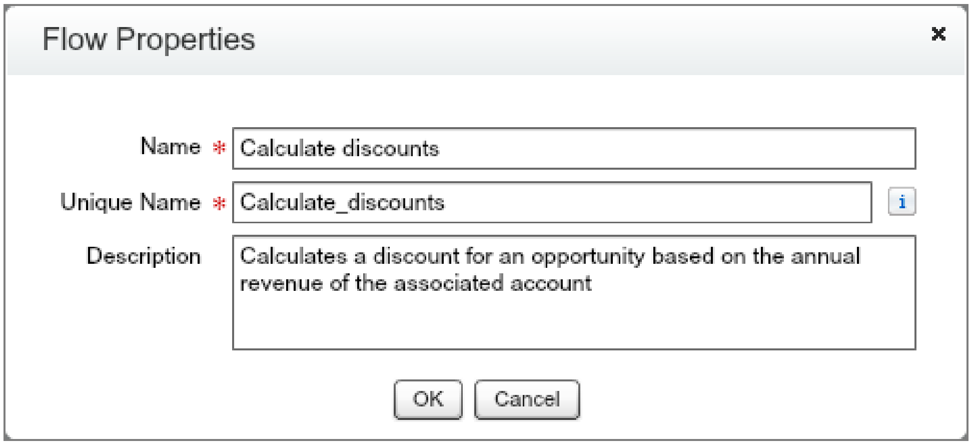
   The Explorer tab lists all of the elements and resources you've added to a flow. It's especially handy if you need to quickly navigate to a particular spot in a large or complex flow.



Step 2: Save the Flow

Once you've added resources to your flow, remember to save. Saving early and often safeguards your flow against unexpected events, such as losing your internet connection, and ensures that you won't have to repeat work that you've already done.

1. On the button bar, click **Save**.
2. For `Name`, type *`Calculate discounts`*.
   The `Unique Name` field is automatically populated based on this entry.
3. For `Description`, type *`Calculates a discount for an opportunity based on the annual revenue of the associated account`*.



Flow Properties

Name * Calculate discounts

Unique Name * Calculate_discounts

Description: Calculates a discount for an opportunity based on the annual revenue of the associated account

4. Click **OK**.
5. Ignore any warnings that appear, and click **OK**.

Step 3: Find the Account Associated with the Opportunity

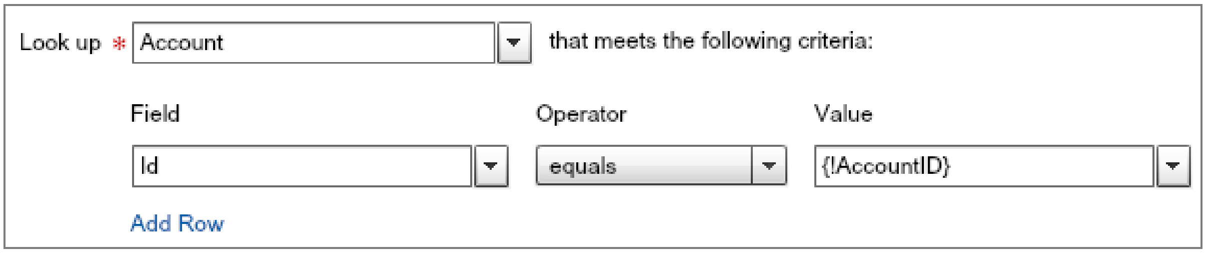
To determine which discount to apply, the flow needs to know the account's annual revenue. The flow has the ID of the account—it's one of the values that was passed into the flow in the first step—so the flow will use this information to look up the annual revenue in Salesforce.

In this step, you'll look up the account using a Record Lookup element. The Cloud Flow Designer includes two elements—Fast Lookup and Record Lookup—that allow you to look up data in Salesforce, but since you're not using an sObject variable to store the account ID, you can't use a Fast Lookup element. In a Record Lookup element, the flow takes two actions. First, the flow looks up a Salesforce record by using criteria you set. In this step, the flow will find an account record whose `Id` field matches the flow variable `{!AccountID}`. Second, the flow assigns one or more field values from the record to variables in the flow. In this step, the flow will assign the value that's stored in the account's `AnnualRevenue` field to the `{!AccountRevenue}` flow variable.

1. From the Palette tab, click and drag a Record Lookup element onto the canvas.
2. For `Name`, type *`Find Related Account`*.
   The `Unique Name` field is automatically populated based on this entry.
3. For `Look up`, click ▼ and select **STANDARD** > **Account**.
   This limits the Record Lookup's search to account records.
4. Set the filter criteria to find the right account.

Field	Value
`Field`	*`Id`*
`Operator`	*`equals`*
`Value`	*`{!AccountID}`*

Here's what the first part of the overlay should look like.



Look up * Account ▼ that meets the following criteria:

Field	Operator	Value
Id ▼	equals ▼	{!AccountID} ▼

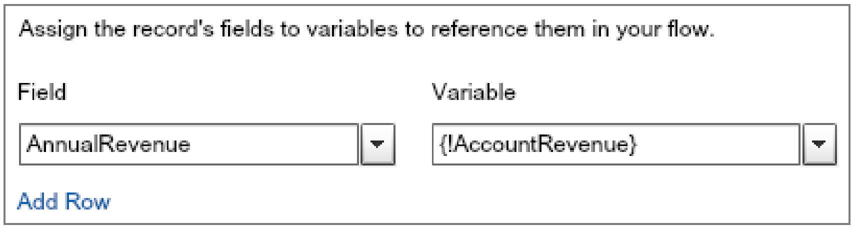
Add Row

5. Configure the element to store the annual revenue in a new variable.
   a. For `Field`, type *`Annual Revenue`*, and select **Annual Revenue** from the picklist.
   b. For `Variable`, click ▼ and select **CREATE NEW** > **Variable**.
   c. Configure the variable.

Field	Value
`Unique Name`	*`AccountRevenue`*
`Description`	*`Stores the revenue (in US dollars) of the account associated with the opportunity`*
`Data Type`	*`Currency`*

Field	Value
`Scale`	*0* Scale is the maximum number of digits allowed to the right of the decimal point.
`Input/Output Type`	*Input and Output*

d. Click **OK** to close the Variable overlay.
Here's what the second part of the overlay should look like.

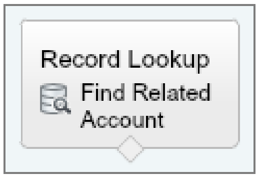


Assign the record's fields to variables to reference them in your flow.

Field	Variable
AnnualRevenue	{!AccountRevenue}

6. Click **OK** to close the Record Lookup overlay.

The flow should now look something like this.



Step 4: Determine the Right Discount to Apply

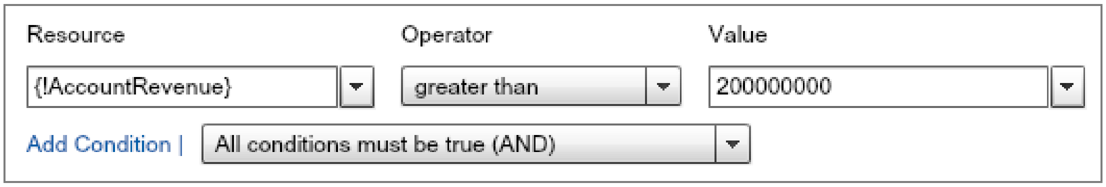
Next, the flow needs to determine which discount to apply. In the new discounting program, Acme Wireless will award discounts based on the annual revenue: the higher the revenue, the larger the discount.

In this step, a Decision element will be used to route the flow along one of several paths, or outcomes, depending on annual revenue. For this tutorial, you'll create two outcomes based on annual revenue: a full discount, which is applied when the account's annual revenue is greater than $200,000,000, and a partial discount, which is applied when the account's annual revenue is between $100,000,000 and $200,000,000. If neither of these conditions are met, the default outcome—no discount—occurs.

1. From the Palette tab, click and drag a Decision element onto the canvas.
2. For `Name`, type *`Determine Discount`*.
   The `Unique Name` field is automatically populated.
3. Create an outcome for the full discount.
   a. Under `Outcomes`, for `Name`, type *`Full Discount`*.
      The `Unique Name` field is automatically.
   b. Set the outcome criteria.

Field	Value
`Resource`	*`{!AccountRevenue}`*
`Operator`	*`greater than`*
`Value`	*`200000000`*

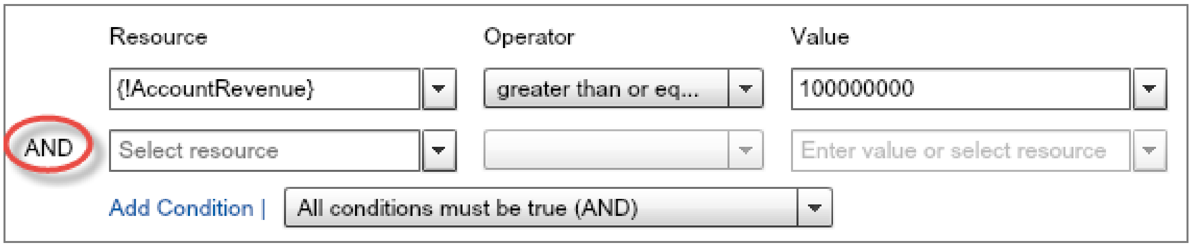
These criteria tell the element to route the flow along the Full Discount path only if the annual revenue of the account is greater than $200,000,000.


Resource: {!AccountRevenue}
Operator: greater than
Value: 200000000

4. Create an outcome for the partial discount. Since this outcome is based on the annual revenue being between two values, you'll use two criteria.
   a. Click **Add Outcome**.
   b. For `Name`, type *`Partial Discount`*.
      The `Unique Name` field is automatically populated based on this entry.
   c. Set the outcome criteria for the lower revenue limit.

Field	Value
`Resource`	*`{!AccountRevenue}`*
`Operator`	*`greater than or equal`*
`Value`	*`100000000`*

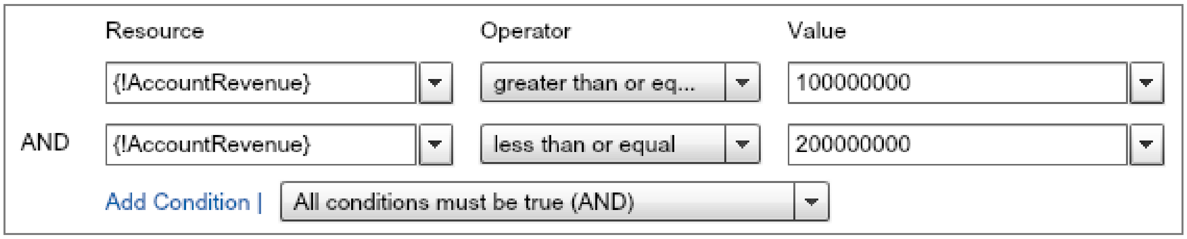
   d. Click **Add Condition**.
      When you add a new condition, the Cloud Flow Designer applies the boolean operator *AND* by default. If you want the flow to take this path, all of the conditions above the operator must be true. Here, *AND* is exactly what you want—a partial discount should only be applied when the annual revenue is above the lower revenue limit *AND* below the upper revenue limit.


Resource: {!AccountRevenue}
Operator: greater than or eq...
Value: 100000000
AND
Select resource
Enter value or select resource
Add Condition |
All conditions must be true (AND)

e. Set the outcome criteria for the upper revenue limit.

Field	Value
`Resource`	*`{!AccountRevenue}`*
`Operator`	*`less than or equal`*
`Value`	*`200000000`*

These criteria tell the element to route the flow along the partial discount path only if the annual revenue of the account is greater than or equal to $100,000,000 and less than or equal to $200,000,000.

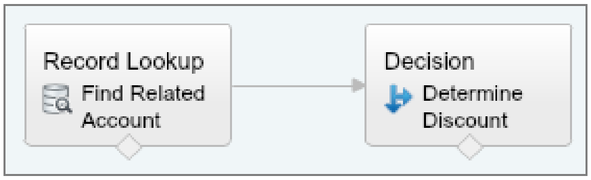


	Resource	Operator	Value
	{!AccountRevenue}	greater than or eq...	100000000
AND	{!AccountRevenue}	less than or equal	200000000

Add Condition | All conditions must be true (AND)

5. Configure the default outcome, which occurs if the account's annual revenue doesn't meet the criteria for a full discount or a partial discount.
   a. Click **[Default Outcome]**.
   b. For `Name`, replace the predefined value with *`No Discount`*.
   c. Click **OK**.
6. Just to be safe, save your progress.
   a. On the button bar, click **Save**.
   b. Ignore any warnings that appear, and click **OK**.
7. Drag the node from the bottom of Find Related Account onto Determine Discount to connect them.

Your flow should now look something like this.



Step 5: Assign Numerical Values to Discounts

At this point, the flow knows the annual revenue of the account and can use that value to direct the flow to the right discount. But the discounts—full, partial, or no—are abstract. To be useful, each discount needs to be linked to a numerical value—10 percent, 20 percent, and so forth.

In the following steps, you'll create a variable in which to store that numerical value and then use Assignment elements to update that variable with the right number based on the outcome. To change a flow variable's value, you'll use the *equals* operator in an Assignment element. The *equals* operator sets a variable to a value, such as a number or string. In the next steps, you'll create a new variable—`Discount`—and use Assignment elements to set the variable's value for the full, partial, and no discount outcomes.

IN THIS SECTION:

Assign a Value for a Full Discount

In this step, you'll create a variable to store the discount percentage. This variable will be used to update the opportunity later in the flow, and its value will differ based on whether it should be a full discount, partial discount, or no discount at all. You'll also assign a 10-percent discount to the variable by using an Assignment element. At the end of the step, you'll connect the Assignment element so that an opportunity receives this discount only if the account meets the criteria for a full discount.

Assign a Value for a Partial Discount

In this step, you'll assign a five-percent discount to the discount variable. At the end of the step, you'll connect the Assignment element so that an opportunity receives this discount only if the account meets the criteria for a partial discount.

Assign a Value for No Discount

In this step, you'll assign a zero percent discount to the discount variable. At the end of the step, you'll connect the Assignment element so that an opportunity receives no discount if the account doesn't meet the criteria for a full or partial discount.

Assign a Value for a Full Discount

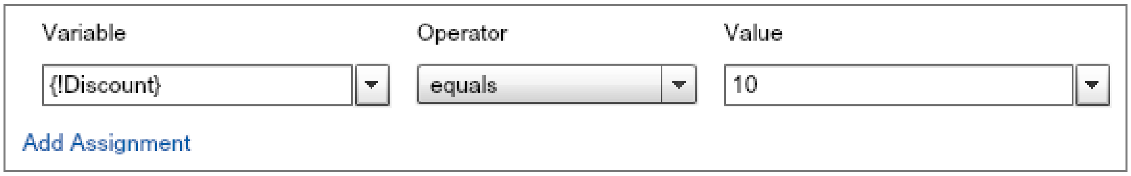
In this step, you'll create a variable to store the discount percentage. This variable will be used to update the opportunity later in the flow, and its value will differ based on whether it should be a full discount, partial discount, or no discount at all. You'll also assign a 10-percent discount to the variable by using an Assignment element. At the end of the step, you'll connect the Assignment element so that an opportunity receives this discount only if the account meets the criteria for a full discount.

1. From the Palette tab, click and drag an Assignment element onto the canvas.
2. For `Name`, type *`Discount is 10%`*.
   The `Unique Name` field is automatically populated based on this entry.
3. Create a variable to store the discount percentage in.
   a. For `Variable`, click ▾ and select **CREATE NEW** > **Variable**.
   b. Configure the variable.

Field	Value
`Unique Name`	*`Discount`*
`Description`	*`Stores the discount percentage in whole number format`*
`Data Type`	*`Number`*
`Scale`	*`2`*
`Input/Output Type`	*`Private`*

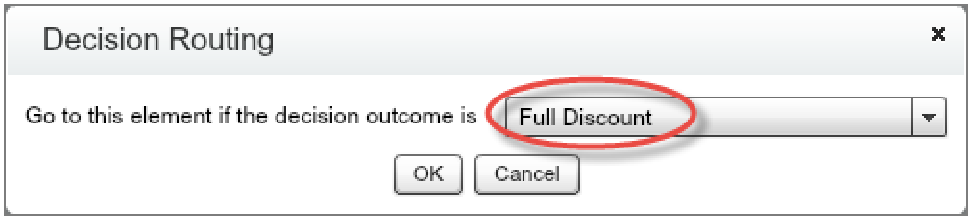
   c. Click **OK**.

4. For `Operator`, select **equals**.
5. For `Value`, type *`10`*.
   Here's what the Assignment overlay should look like.



Variable	Operator	Value
{!Discount}	equals	10

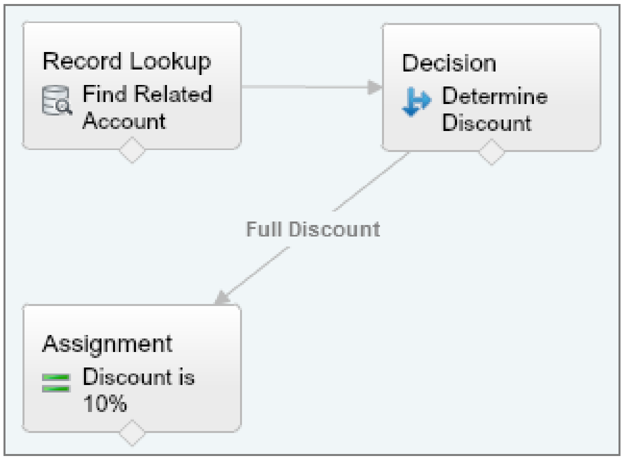
6. Click **OK**.
7. Drag the node from the bottom of Determine Discount onto Discount is 10% to connect them.
8. Verify that the Decision Routing overlay is set to **Full Discount**.



Decision Routing

Go to this element if the decision outcome is Full Discount

Your flow should now look something like this.



Assign a Value for a Partial Discount

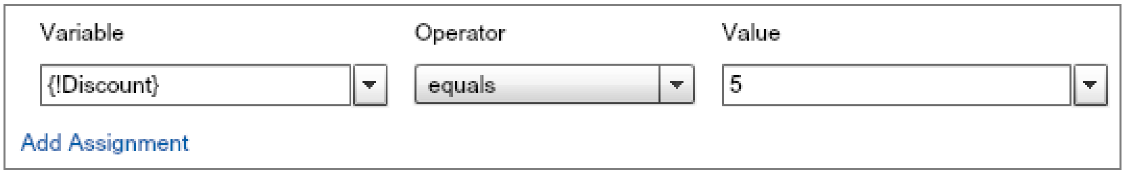
In this step, you'll assign a five-percent discount to the discount variable. At the end of the step, you'll connect the Assignment element so that an opportunity receives this discount only if the account meets the criteria for a partial discount.

1. From the Palette tab, click and drag an Assignment element onto the canvas.
2. For `Name`, type *`Discount is 5%`*.
   The `Unique Name` field is automatically populated based on this entry.

3. Set the assignment fields.

Field	Value
`Variable`	*`{!Discount}`*
`Operator`	*`equals`*
`Value`	*`5`*

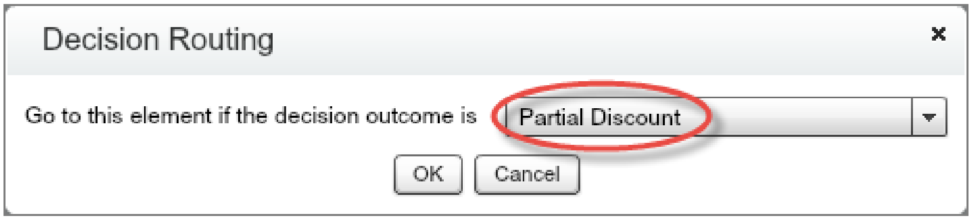
Here's what the Assignment overlay should look like.



Variable	Operator	Value
{!Discount}	equals	5

Add Assignment

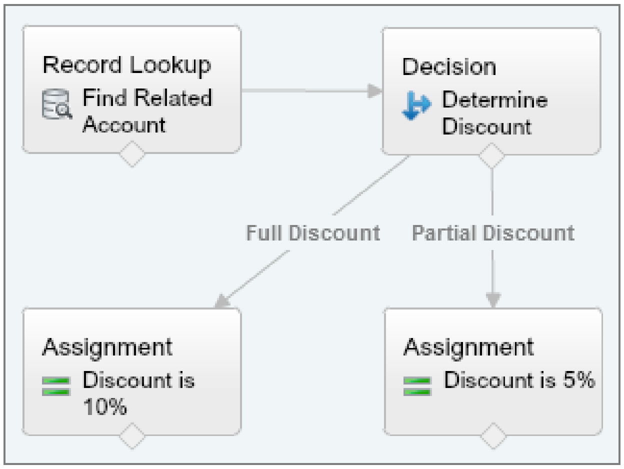
4. Click **OK**.
5. Drag the node from the bottom of Determine Discount onto Discount is 5% to connect them.
6. Verify that the Decision Routing overlay is set to **Partial Discount**.



Decision Routing

Go to this element if the decision outcome is Partial Discount

Your flow should now look something like this.



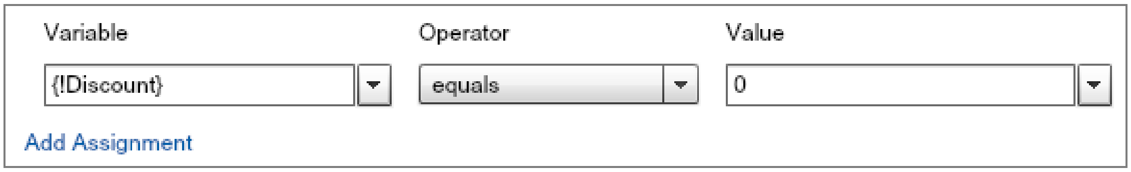
Assign a Value for No Discount

In this step, you'll assign a zero percent discount to the discount variable. At the end of the step, you'll connect the Assignment element so that an opportunity receives no discount if the account doesn't meet the criteria for a full or partial discount.

1. From the Palette tab, click and drag an Assignment element onto the canvas.
2. For `Name`, type *`Discount is 0%`*.
   The `Unique Name` field is automatically populated based on this entry.
3. Set the assignment fields.

Field	Value
`Variable`	*`{!Discount}`*
`Operator`	*`equals`*
`Value`	*`0`*

Here's what the Assignment overlay should look like.



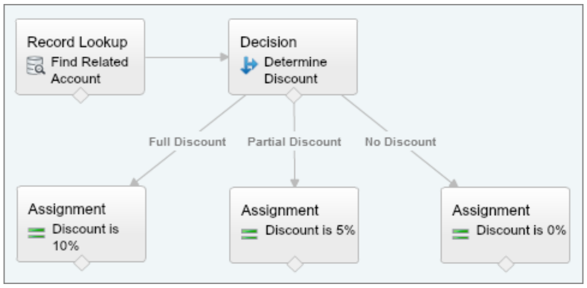
Variable: {!Discount}

Operator: equals

Value: 0

4. Click **OK**.
5. Drag the node from the bottom of Determine Discount onto Discount is 0% to connect them.
   Because you've already connected the Full and Partial Discount outcomes to Assignment elements, the Cloud Flow Designer assigns the only remaining outcome—No Discount—to this path by default.
6. Just to be safe, save your progress.
   a. On the button bar, click **Save**.
   b. Ignore any warnings that appear and click **OK**.

Your flow should now look something like this.



Step 6: Update the Opportunity with the Discount

Once the flow has determined which discount to apply, it needs to return that information back to the opportunity so that the sales associate knows how much of a discount to offer the customer.

In this step, you'll update the opportunity with the value of the discount variable by using a Record Update element. In a Record Update element, the flow takes two actions. First, the flow looks up a Salesforce record by using criteria you set. In this step, the flow will find an opportunity record whose `Id` matches the flow variable `{!OpportunityID}`, which stores the ID of the opportunity that triggered the flow. Second, the flow updates one or more of the record's fields. The Record Update element updates all of the records that meet the criteria that you set, but because IDs are unique, the flow will only update one opportunity. In this step, the flow will update the opportunity's `Discount Percentage` field with the value of `{!Discount}`. `Discount Percentage` is one of the two custom fields included in the enhanced Warehouse data model. The field appears on the opportunity page layout and displays the discount in percentage format. It's also used to calculate the other custom field, `Discounted Amount`. `Discounted Amount` displays the value of the opportunity after any discount has been applied. Salesforce updates this field automatically once the `Discount Percentage` field is populated by the flow.

1. From the Palette tab, click and drag a Record Update element onto the canvas.

   Note: If you're wondering why you can't use a Fast Update element here, remember that you're storing the discount in a variable and not an sObject variable. Fast Update elements require sObject variables or sObject collection variables.

2. For `Name`, type *`Update Discount Field`*.
   The `Unique Name` field is automatically populated based on this entry.

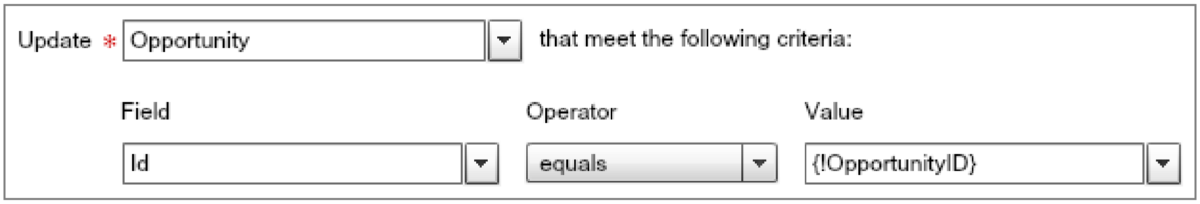
3. For `Update`, type *`Opportunity`*, and select **Opportunity** from the picklist.

   This tells the Record Update element to limit its search to Opportunity records.

4. Configure the filter criteria to find the opportunity that triggered the flow.

Field	Value
`Field`	*`Id`*
`Operator`	*`equals`*
`Value`	*`{!OpportunityID}`*

Here's what the overlay should look like.



Update * Opportunity that meet the following criteria:

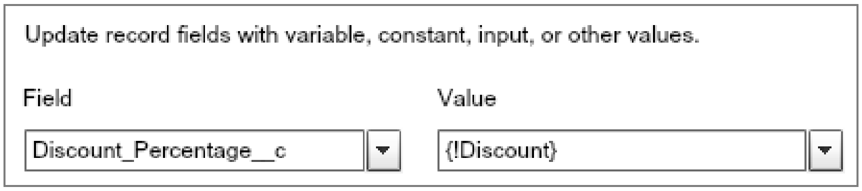
Field	Operator	Value
Id	equals	{!OpportunityID}

5. Configure the element to update the `Discount Percentage` field in the opportunity with the value stored in the flow variable `Discount`.

Field	Value
`Field`	*`Discount_Percentage__c`*

Field	Value
`Value`	*`{!Discount}`*

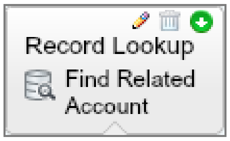
Here's what the overlay should look like now.



Update record fields with variable, constant, input, or other values.

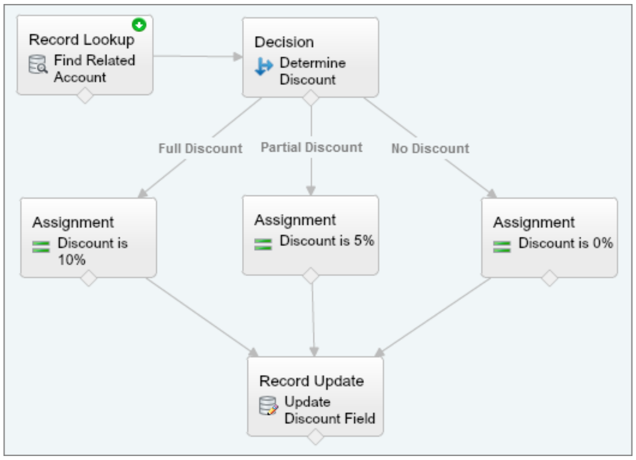
Field	Value
Discount_Percentage__c ▾	{!Discount} ▾

6. Click **OK**.
7. Connect the three Assignment elements to the Record Update element so that the opportunity is updated no matter which outcome was executed.
   a. Drag the node from the bottom of Discount is 10% onto Update Discount Field.
   b. Drag the node from the bottom of Discount is 5% onto Update Discount Field.
   c. Drag the node from the bottom of Discount is 0% onto Update Discount Field.
8. Designate a start element for the flow.
   a. Hover your mouse over Find Related Account and three icons appear in the upper right corner.



   b. Click ⊕ to set Find Related Account as the start element.
9. Click **Save**.

Your finished flow should look something like this.

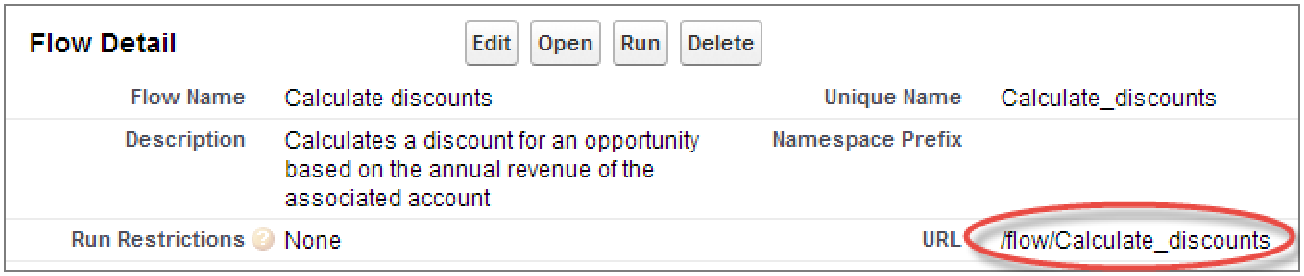


Step 7: Implement the Flow with a Custom Button

You've successfully created the flow, but it's only accessible from Setup. Let's implement the flow so that your sales representatives can easily find and use it.

In this step, you'll create a custom button that will be displayed on opportunities in Salesforce. When sales associates click the button, the flow calculates a discount for the opportunity and updates it.

1. Get the URL of the flow.
   a. On the button bar, click **Close**.
   b. From the flow detail page, copy the flow's URL.



Flow Detail

Flow Name	Calculate discounts	Unique Name	Calculate_discounts
Description	Calculates a discount for an opportunity based on the annual revenue of the associated account	Namespace Prefix	
Run Restrictions	None	URL	/flow/Calculate_discounts

2. Create the custom button.
   a. From the object management settings for opportunities, go to Buttons, Links, and Actions.
   b. Click **New Button or Link**.

3. Configure how the custom button appears.

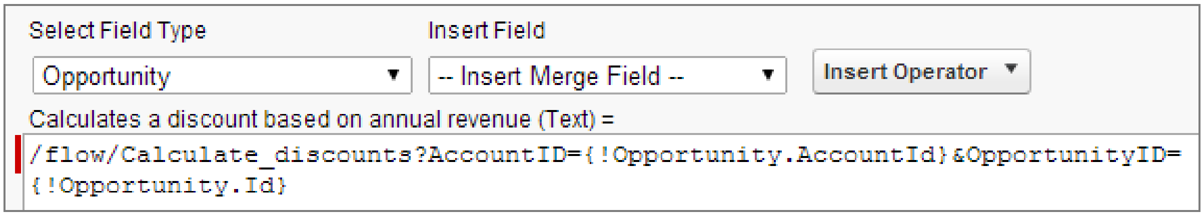
Field	Value
`Label`	*`Calculate Discount`*
`Name`	*`Calculate_Discount_button`*
`Description`	*`Calculates a discount based on annual revenue`*
`Display Type`	*`Detail Page Button`*
`Behavior`	*`Display in new window`*
`Content Source`	*`URL`*

4. Link the button to the flow using URL parameters.
   a. In the text box, paste the flow URL.
   b. At the end of the URL, append the following.

```
?AccountID={!Opportunity.AccountId}&OpportunityID={!Opportunity.Id}
```

Now, when the flow is called, two field values—`AccountId` and `Id`—are passed into the flow from the opportunity. The ID of the account that's associated with the opportunity, `AccountId`, gets assigned to the flow variable `{!AccountID}`. The ID of the opportunity itself, `Id`, gets assigned to the flow variable `{!OpportunityID}`.

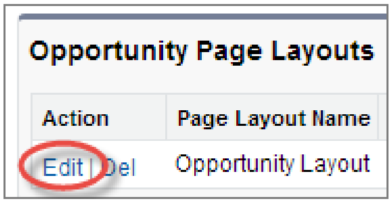
Here's what the URL should look like.



```
/flow/Calculate_discounts?AccountID={!Opportunity.AccountId}&OpportunityID=
{!Opportunity.Id}
```

c. Click **Check Syntax** to verify that you've entered a valid URL.

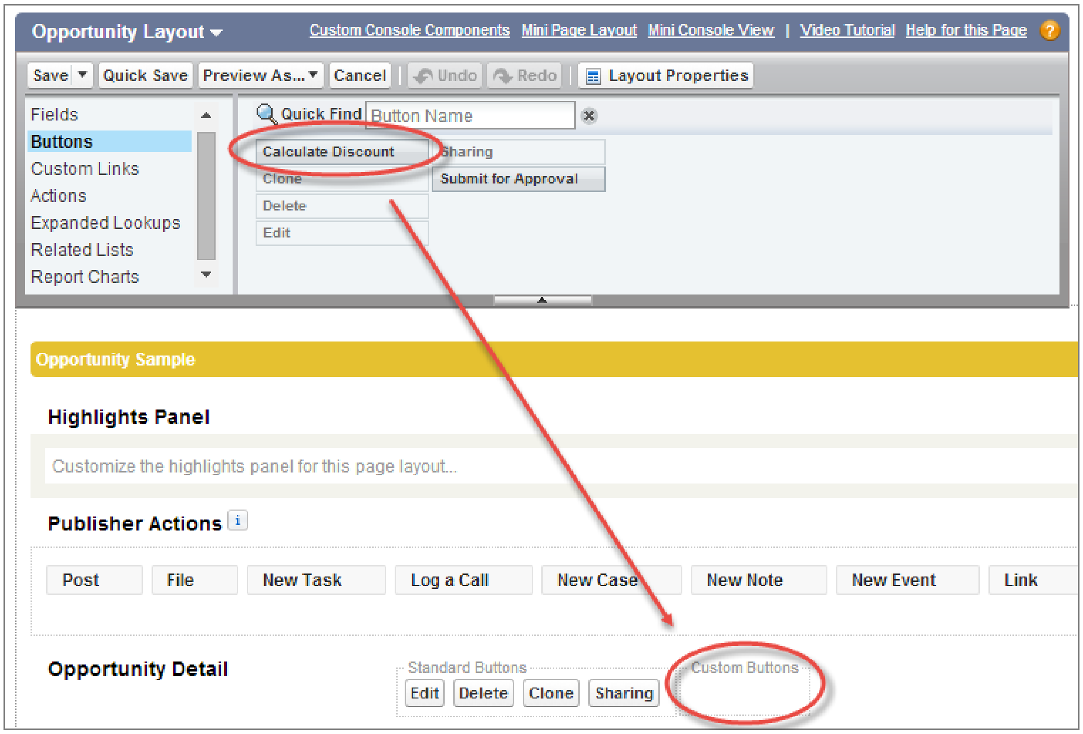
5. Click **Save**.
   A dialog box appears to remind you to add the custom button to the opportunity page layout.
6. Click **OK**.
7. Add the button to the opportunity page layout, so that it appears on all opportunity records.
   a. From the object management settings for opportunities, go to Page Layouts.
   b. In the Action column next to Opportunity Layout, click **Edit**.



Opportunity Page Layouts

Action	Page Layout Name
Edit | Del	Opportunity Layout

   c. In the left column, click **Buttons**.
   d. Click and drag **Calculate Discount** onto Custom Buttons.



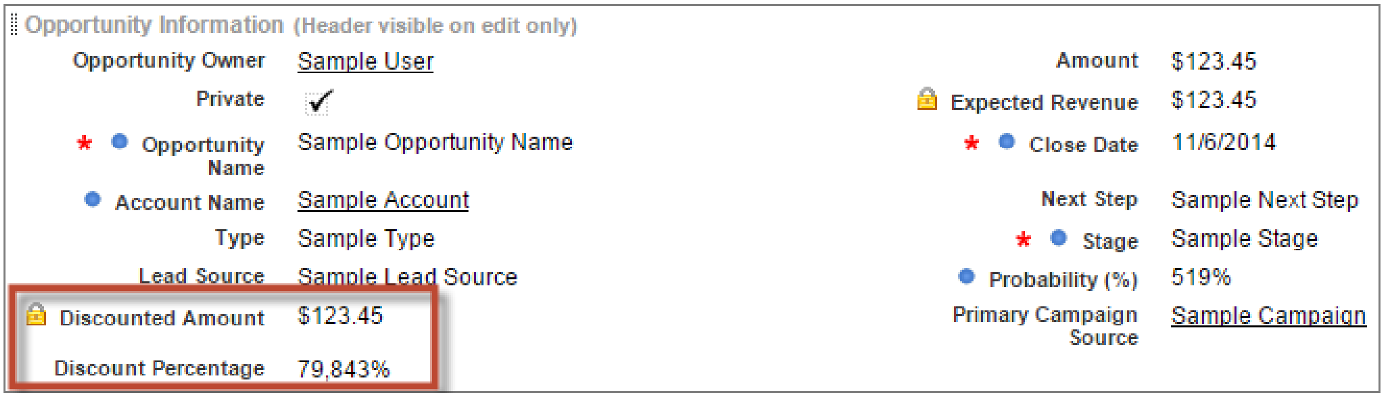
e. Click **Save**.
Now, when sales associates visit an opportunity page in Salesforce they'll see a Calculate Discount button at the top of the page.

Step 8: Customize the Opportunity Page Layout

Soon, you'll want to test the flow—just to make sure that everything works. Before you can do that, add the `Discount Percentage` and `Discounted Amount` fields to the opportunity page layout.

Those fields were included in the enhanced Warehouse data model that you installed earlier, but they weren't added to the page layout automatically.

1. From the object management settings for opportunities, go to Page Layouts.
2. Next to Opportunity Layout, click **Edit**.
3. From the palette, drag the `Discounted Amount` and `Discount Percentage` fields into the Opportunity Information section.
4. Verify that the page layout looks like this, and then click **Save**.



Opportunity Information (Header visible on edit only)

Opportunity Owner	Sample User	Amount	$123.45
Private	✓	Expected Revenue	$123.45
Opportunity Name	Sample Opportunity Name	Close Date	11/6/2014
Account Name	Sample Account	Next Step	Sample Next Step
Type	Sample Type	Stage	Sample Stage
Lead Source	Sample Lead Source	Probability (%)	519%
Discounted Amount	$123.45	Primary Campaign Source	Sample Campaign
Discount Percentage	79.843%		

Step 9: Create a Sample Opportunity

Just one more thing before you can test the flow. Let's create a sample opportunity so that the flow has something to update.

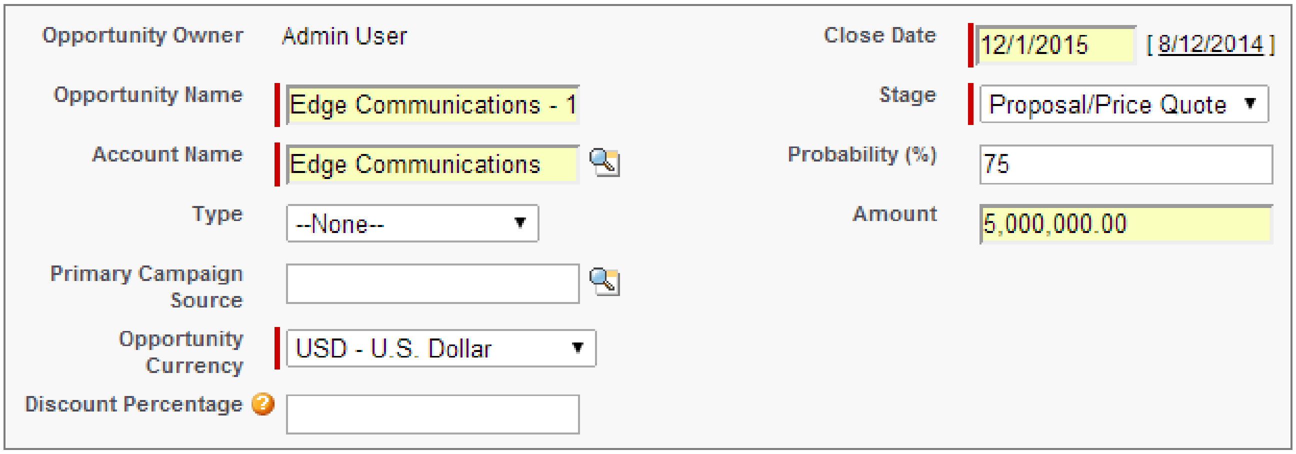
The sample opportunity is for a one-year service contract with Edge Communications, a Texas-based electronics company with $139,000,000 in annual revenue.

1. Click the **Opportunities** tab.
2. Click **New**.

   If prompted to choose an opportunity record type, select **B2B Sales** and then click **Continue**.
3. Configure the opportunity with the following values.

Field	Value
`Opportunity Name`	*`Edge Communications - 1 year contract`*
`Account Name`	*`Edge Communications`*
`Close Date`	*`12/1/2015`*
`Stage`	*`Proposal/Price Quote`* The `Probability` field is automatically populated based on this entry.
`Amount`	*`5,000,000`*

4. Verify that the opportunity looks like this, and then click **Save**.



Opportunity Owner	Admin User	Close Date	12/1/2015 [8/12/2014]
Opportunity Name	Edge Communications - 1	Stage	Proposal/Price Quote
Account Name	Edge Communications	Probability (%)	75
Type	--None--	Amount	5,000,000.00
Primary Campaign Source			
Opportunity Currency	USD - U.S. Dollar		
Discount Percentage			

Step 10: Test the Flow

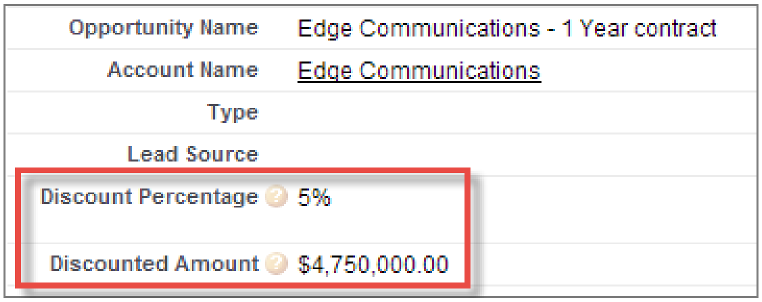
Testing is an important part of building a flow, allowing you to fine-tune the flow's behavior, identify and fix bugs, and otherwise make sure your users have a pleasant experience. And of course, you benefit, since you're much less likely to spend time later responding to support requests and fixing errors.

In this example, you'll calculate the discount for the opportunity that you created in the last step.

1. In the global search bar, search for `Edge Communications - 1 year contract`.
2. In the search results, click **Edge Communications - 1 year contract**.

   The amount, or value of the contract, is $5,000,000. Because the annual revenue of Edge Communications exceeds $100,000,000 but is less than $200,000,000, the flow should return a 5% discount ($250,000). Thus, the discounted amount for this opportunity should be $4,750,000. Let's verify that the discount is calculated correctly.

3. Click **Calculate Discount**.
   A window appears notifying you that the flow has finished running.
4. Close the window.
5. Refresh the opportunity page to see the updated field values.
6. Verify that `Discount Percentage` has a value of 5% and `Discounted Amount` has a value of $4,750,000.



Opportunity Name	Edge Communications - 1 Year contract
Account Name	Edge Communications
Type	
Lead Source	
Discount Percentage	5%
Discounted Amount	$4,750,000.00

There are a number of ways to extend the flow you created in this tutorial. If, at some point, Acme Wireless decides to limit the number of full discounts it offers, you could configure the flow to require approval from a manager before it updates the record. On the other hand, if Acme decides it wants to be more aggressive with its discount program, you could configure the flow to look up whether the last opportunity was lost on cost and, if so, offer the customer an even bigger discount.

Reassign Accounts

As one of the fastest growing companies in the segment, Acme Wireless' employees frequently move on to new, expanded roles within their teams. When this happens, the manager reassigns an employee's accounts to teammates based on the account's rating—hot, warm, or cold—and lets the teammates know that they've inherited accounts. To save managers' time, you've been asked to automate this process.

In this tutorial, you'll build a flow that allows managers to easily reassign account ownership when an employee moves on to a new role at the company. Then, you'll implement the flow using a Visualforce tab that will appear at the top of the Salesforce app of select users at Acme Wireless.

Before you dive into this tutorial, make sure that you have:

- Signed up for a Developer Edition organization at developer.salesforce.com/signup.
- Installed the enhanced Warehouse data model.

IN THIS SECTION:

Step 10: Test the Flow

Testing is an important part of building a flow, allowing you to fine-tune the flow's behavior, identify and fix bugs, and otherwise make sure your users have a pleasant experience. And of course, you benefit, since you're much less likely to spend time later responding to support requests and fixing errors.

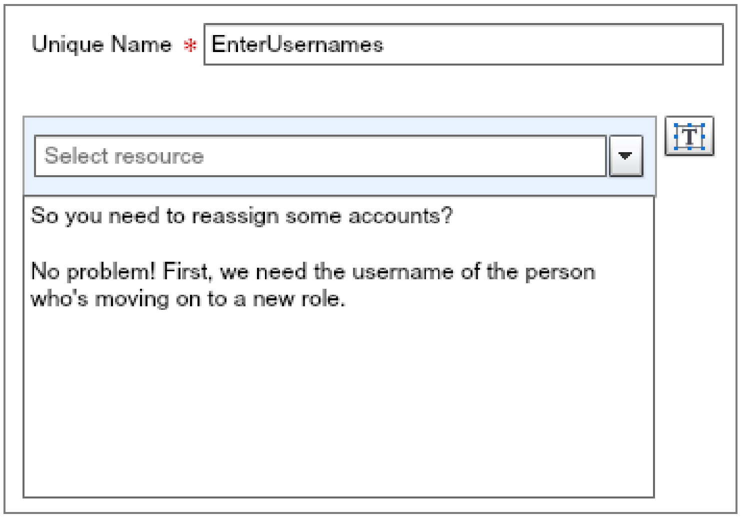
Step 1: Collect Email Addresses

Before the flow can reassign accounts, it needs to know who's involved. Since an Acme Wireless manager knows which employee is moving on to the new role and which teammates are getting new accounts, you'll need to set up the flow to collect this information.

In this step, you'll configure a Screen element where the manager can identify the employees whose account assignments are changing.

1. From Setup, enter `Flows` in the `Quick Find` box, then select **Flows**.
2. Click **New Flow**.
3. From the Palette tab in the left pane, click and drag a Screen element onto the canvas.
4. For `Name`, type `Collect Usernames`.
   The `Unique Name` field is automatically populated based on this entry.
5. Welcome the manager to the flow.
   a. From the Add a Field tab, double-click **Display Text**.
   b. Click **[Display Text]** in the preview pane.
   c. Configure the field.

Field	Value
`Unique Name`	*EnterUsernames*
text box	*So you need to reassign some accounts?* *No problem! First, we need the username of the person who's moving on to a new role.*

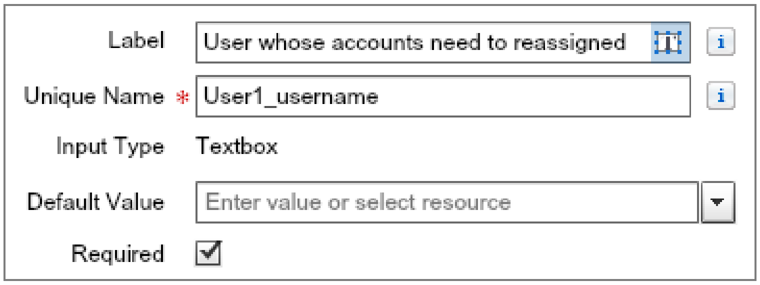

Unique Name * EnterUsernames

So you need to reassign some accounts?

No problem! First, we need the username of the person who's moving on to a new role.

6. Add a text field where the manager can enter the username of the person who's moving on to a new role.
   a. From the Add a Field tab, double-click **Textbox**.
   b. Click **[Textbox]** in the preview pane.
   c. Configure the field.

Field	Value
`Label`	*`User whose accounts need to be reassigned`*
`Unique Name`	*`User1_username`*
`Required`	Selected



Label User whose accounts need to reassigned

Unique Name * User1_username

Input Type Textbox

Default Value Enter value or select resource

Required ☑

7. Prompt the manager to enter the other usernames.
   a. From the Add a Field tab, double-click **Display Text**.
   b. Click **[Display Text]** in the preview pane.
   c. Configure the field.

Field	Value
`Unique Name`	*`other_usernames`*
text box	*`Then we need the usernames of the people we want to give the accounts to.`*

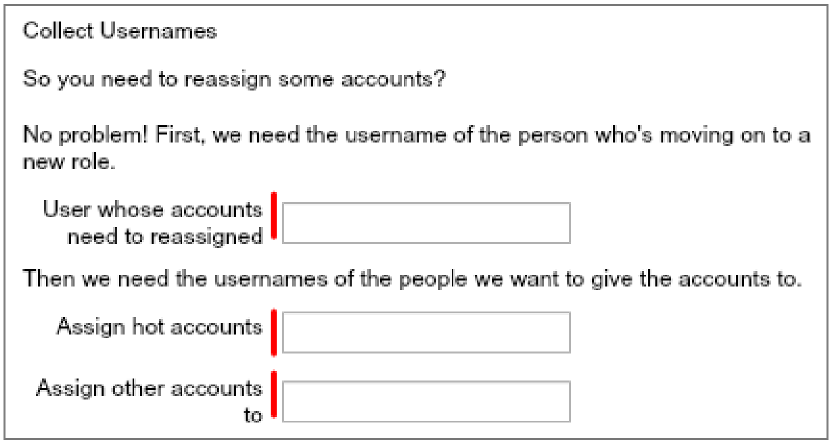
8. Add a text field where the manager can enter the username of the employee who's getting the hot accounts.
   a. From the Add a Field tab, double-click **Textbox**.
   b. Click **[Textbox]** in the preview pane.
   c. Configure the field.

Field	Value
Label	*Assign hot accounts to*
Unique Name	*User2_username*
Required	Selected

9. Add a text field where the manager can enter the username of the employee who's getting the warm and cold accounts.
   a. From the Add a Field tab, double-click **Textbox**.
   b. Click **[Textbox]** in the preview pane.
   c. Configure the field.

Field	Value
Label	*Assign other accounts to*
Unique Name	*User3_username*
Required	Selected

10. Verify that your preview pane looks like this, and then click **OK**.


Collect Usernames

So you need to reassign some accounts?

No problem! First, we need the username of the person who's moving on to a new role.

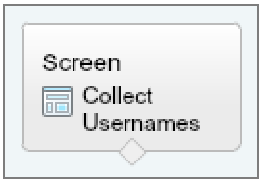
User whose accounts need to reassigned

Then we need the usernames of the people we want to give the accounts to.

Assign hot accounts

Assign other accounts to

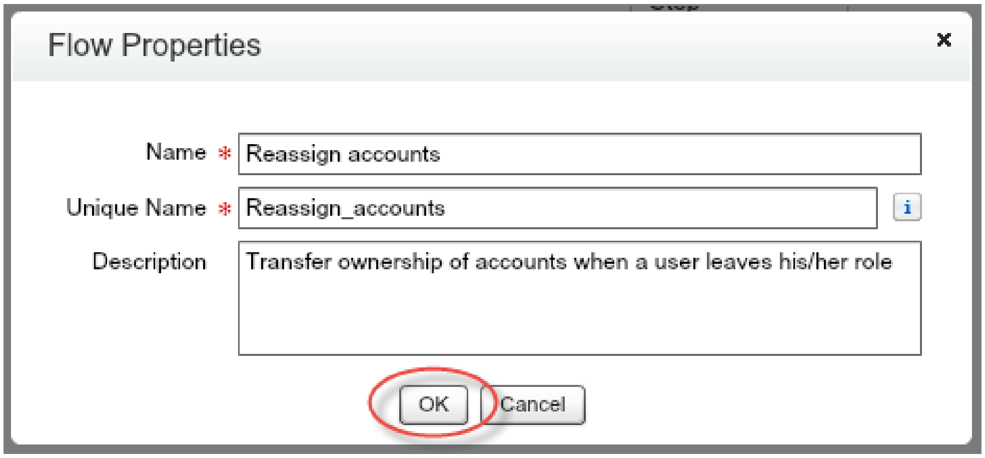
Your flow should now look something like this.



Step 2: Save the Flow

Once you've configured the first element in your flow, remember to save. Saving early and often safeguards your flow against unexpected events, such as losing your internet connection, and ensures that you won't have to repeat work that you've already done.

1. On the button bar, click **Save**.
2. For `Name`, type *`Reassign accounts`*.
   The `Unique Name` field is automatically populated based on this entry.
3. For `Description`, type *`Transfer ownership of accounts when a user leaves his/her role`*.
4. Click **OK**.



Flow Properties

Name * Reassign accounts

Unique Name * Reassign_accounts

Description Transfer ownership of accounts when a user leaves his/her role

5. Ignore any warnings that appear, and click **OK**.

Step 3: Match Usernames to Salesforce Users

Next, the flow needs to match the usernames that the manager entered to user records in Salesforce. Later on, the flow will use this information to look up accounts and reassign them to the right employees.

In the following steps, you'll use a series of Fast Lookup elements to match usernames with users. In a Fast Lookup element, the flow takes two actions. First, the flow looks up a Salesforce record by using criteria that you set. To find the first employee, the flow searches for a user record whose `Username` field matches the username that the manager typed into the `User1_username` textbox on the first screen of the flow. Second, the flow assigns one or more field values from the record to an sObject variable in the flow. In the case of the first employee, the flow assigns the values that are stored in the user's `Id`, `FirstName`, `LastName`, and `Email` fields to the `{!User1}` sObject variable.

Tip: You may be wondering if a Record Lookup element would work here. Definitely—but it wouldn't be as efficient as a Fast Lookup element. Here's why: in a Record Lookup, you need to identify a variable for each field value that you want to store, even if every field value is being stored in the same sObject variable. For this flow, that means identifying one variable for `Id`, two for `FirstName` and `LastName`, and one more for `Email`. In a Fast Lookup, you can identify an sObject variable just once, and the flow knows to store every field value in that sObject variable.

IN THIS SECTION:

Look Up the Employee Who's Taking Over the Hot Accounts

The flow located the employee who's moving on to the new role and then stored the user's ID, name, and email address in an sObject variable. Now, repeat the process for the employee who's being assigned the hot accounts. As before, you'll do this with a Fast Lookup element.

Look Up the Employee Who's Taking Over the Warm and Cold Accounts

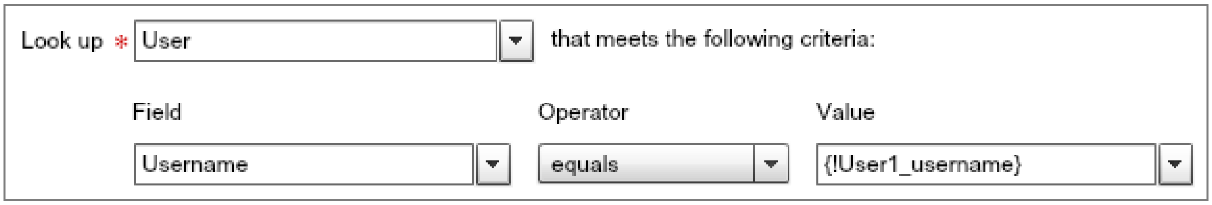
There's still one employee left for the flow to look up: the person who's being assigned the warm and cold accounts. Just like in the last step, you'll use a Fast Lookup to find the employee's user record in Salesforce and then store some field values to use later in the flow.

Look Up the Employee Who's Moving on to a New Role

In this step, the flow finds the employee who's moving on to the new role by using a Fast Lookup element.

1. From the Palette tab in the left pane, click and drag a Fast Lookup element onto the canvas.
2. For Name, type *Get User 1's ID*.
   The Unique Name field is automatically populated based on this entry.
3. For Look up, type *User*, and select **User** from the picklist.
4. Set the filter criteria to match the username of the first employee to the right user record in Salesforce.

Field	Value
Field	*Username*
Operator	*equals*
Value	*{!User1_username}*



Look up * User ▼ that meets the following criteria:

Field	Operator	Value
Username ▼	equals ▼	{!User1_username} ▼

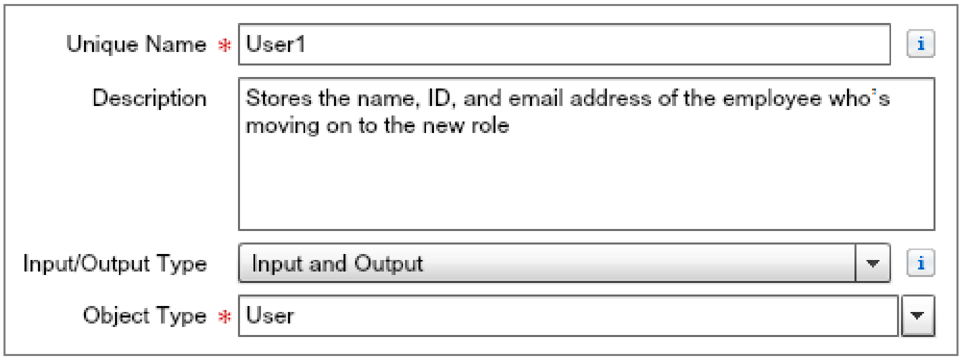
5. You'll need the employee's ID to look up and reassign accounts, so create an sObject variable to store the Id field value from this user record.

   Later in the flow, you'll use the employee's name and email address to confirm whose accounts were updated, so you'll have the variable store the FirstName, LastName, and Email fields, too.

   a. For the Variable field, click ▼ and select **CREATE NEW** > **SObject Variable**.
   b. Configure the sObject variable.

Field	Value
Unique Name	*User1*

Field	Value
`Description`	*Stores the name, ID, and email address of the employee who’s moving on to the new role*
`Input/Output Type`	*Input and Output*
`Object Type`	*User*



Unique Name *	User1
Description	Stores the name, ID, and email address of the employee who's moving on to the new role
Input/Output Type	Input and Output
Object Type *	User

c. Click **OK**.

6. Configure the variable to store the `Id`, `FirstName LastName`, and `Email` field values from the record.

   The Cloud Flow Designer gives you a row to store the first field by default. If you want to store more fields, you’ll need to add rows. Since you have a total of four fields to store, add three rows.

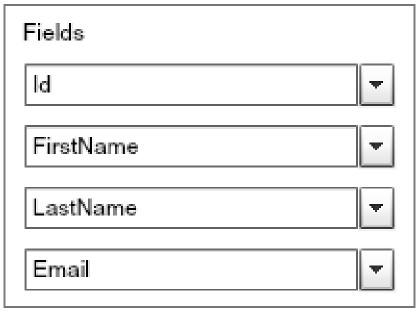
   a. Click add **Add Row** three times.

   b. For `Fields`, type *`Id`*, and select **Id** from the picklist.

   c. In the next row, type *`FirstName`* and select **FirstName** from the picklist.

   d. In the next row, type *`LastName`* and select **LastName** from the picklist.

   e. In the last row, type *`Email`* and select **Email** from the picklist.



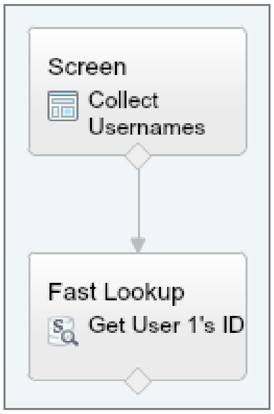
Fields

- Id
- FirstName
- LastName
- Email

7. Click **OK**.

8. Drag the node from the bottom of Collect Usernames onto Get User 1’s ID to connect the elements.

Your flow should now look something like this.



Look Up the Employee Who’s Taking Over the Hot Accounts

The flow located the employee who’s moving on to the new role and then stored the user’s ID, name, and email address in an sObject variable. Now, repeat the process for the employee who’s being assigned the hot accounts. As before, you'll do this with a Fast Lookup element.

1. From the Palette tab in the left pane, click and drag a Fast Lookup element onto the canvas.
2. For `Name`, type *`Get User 2's ID`*.
   The `Unique Name` field is automatically populated based on this entry.
3. For `Look up`, type *`User`*, and select **User** from the picklist.
4. Set the filter criteria to match the username of the second employee to the right user record in Salesforce.

Field	Value
`Field`	*`Username`*
`Operator`	*`equals`*
`Value`	*`{!User2_username}`*

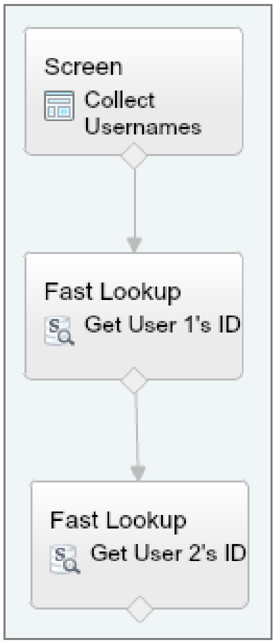
5. Create a new sObject variable to store the `Id`, `FirstName LastName`, and `Email` fields from the record.
   a. For the `Variable` field, click ▾ and select **CREATE NEW** > **SObject Variable**.
   b. Enter the following values.

Field	Value
`Unique Name`	*`User2`*
`Description`	*`Stores the name, ID, and email address of the employee who’s being assigned the hot accounts`*
`Input/Output Type`	*`Input and Output`*
`Object Type`	*`User`*

   c. Click **OK**.

6. Store the `Id`, `FirstName LastName`, and `Email` field values in the sObject variable.
   a. Click **Add Row** three times.
   b. For `Fields`, type *`Id`*, and select **Id** from the picklist.
   c. In the next row, type *`FirstName`* and select **FirstName** from the picklist.
   d. In the next row, type *`LastName`* and select **LastName** from the picklist.
   e. In the last row, type *`Email`* and select **Email** from the picklist.
7. Click **OK**.
8. Drag the node from the bottom of Get User 1's ID onto Get User 2's ID to connect the elements.

Your flow should now look something like this.



Look Up the Employee Who's Taking Over the Warm and Cold Accounts

There's still one employee left for the flow to look up: the person who's being assigned the warm and cold accounts. Just like in the last step, you'll use a Fast Lookup to find the employee's user record in Salesforce and then store some field values to use later in the flow.

1. From the Palette tab in the left pane, click and drag a Fast Lookup element onto the canvas.
2. For `Name`, type *`Get User 3's ID`*.
   The `Unique Name` field is automatically populated based on this entry.
3. For `Look up`, type *`User`*, and select **User** from the picklist.
4. Set the filter criteria to match the username of the third employee to the right user record in Salesforce.

Field	Value
`Field`	*`Username`*
`Operator`	*`equals`*
`Value`	*`{!User3_username}`*

5. Create a new sObject variable to store the `Id`, `FirstName`, `LastName`, and `Email` fields from the record.
   a. For the `Variable` field, click ▼ and select **CREATE NEW** > **SObject Variable**.
   b. Enter the following values.

Field	Value
`Unique Name`	*`User3`*
`Description`	*`Stores the name, ID, and email address of the employee who’s being assigned the warm and cold accounts`*
`Input/Output Type`	*`Input and Output`*
`Object Type`	*`User`*

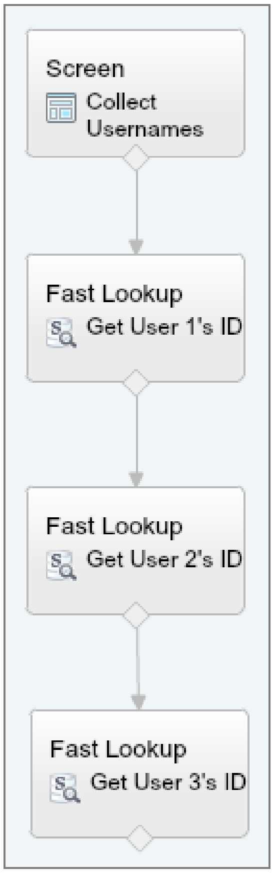
   c. Click **OK**.

6. Configure the variable to store the `Id`, `FirstName`, `LastName`, `Email` field values from the record.
   a. Click **Add Row** twice.
   b. For `Fields`, type *`Id`*, and select **Id** from the picklist.
   c. In the next row, type *`FirstName`* and select **FirstName** from the picklist.
   d. In the next row, type *`LastName`* and select **LastName** from the picklist.
   e. In the last row, type *`Email`* and select **Email** from the picklist.

7. Click **OK**.

8. Drag the node from the bottom of Get User 2’s ID onto Get User 3’s ID to connect the elements.

Your flow should now look something like this.



Step 4: Find the Accounts You Want to Reassign

After identifying which Salesforce user is moving on to the new role and which teammates are getting new accounts, the flow needs to look up the right accounts to reassign.

To find them, the flow will use the ID of the first employee—one of the field values you've conveniently stored in the `{!User1}` sObject variable. Just like before, you'll use a Fast Lookup element. Now, instead of storing field values in an sObject variable, you'll use an sObject collection variable. Like an sObject variable, an sObject collection variable can store multiple field values. But where an sObject variable is limited to storing values from a single record, an sObject collection variable can store values from every record that meets the filter criteria that you set. Once they've been added to an sObject collection variable, these records are called *items*. Since the employee who's moving on to the new role has more than one account to reassign, you need an sObject collection variable.

1. From the Palette tab in the left pane, click and drag a Fast Lookup element onto the canvas.
2. For `Name`, type *`Find User 1's Accounts`*.
   The `Unique Name` field is automatically populated based on this entry.
3. Configure the Fast Lookup element to find the accounts that are owned by the employee who's moving on to the new role.
   a. For `Look up`, type *`Account`*, and select **Account** from the picklist.
   b. Set the filter criteria to match the ID of the first employee to the right account records in Salesforce.

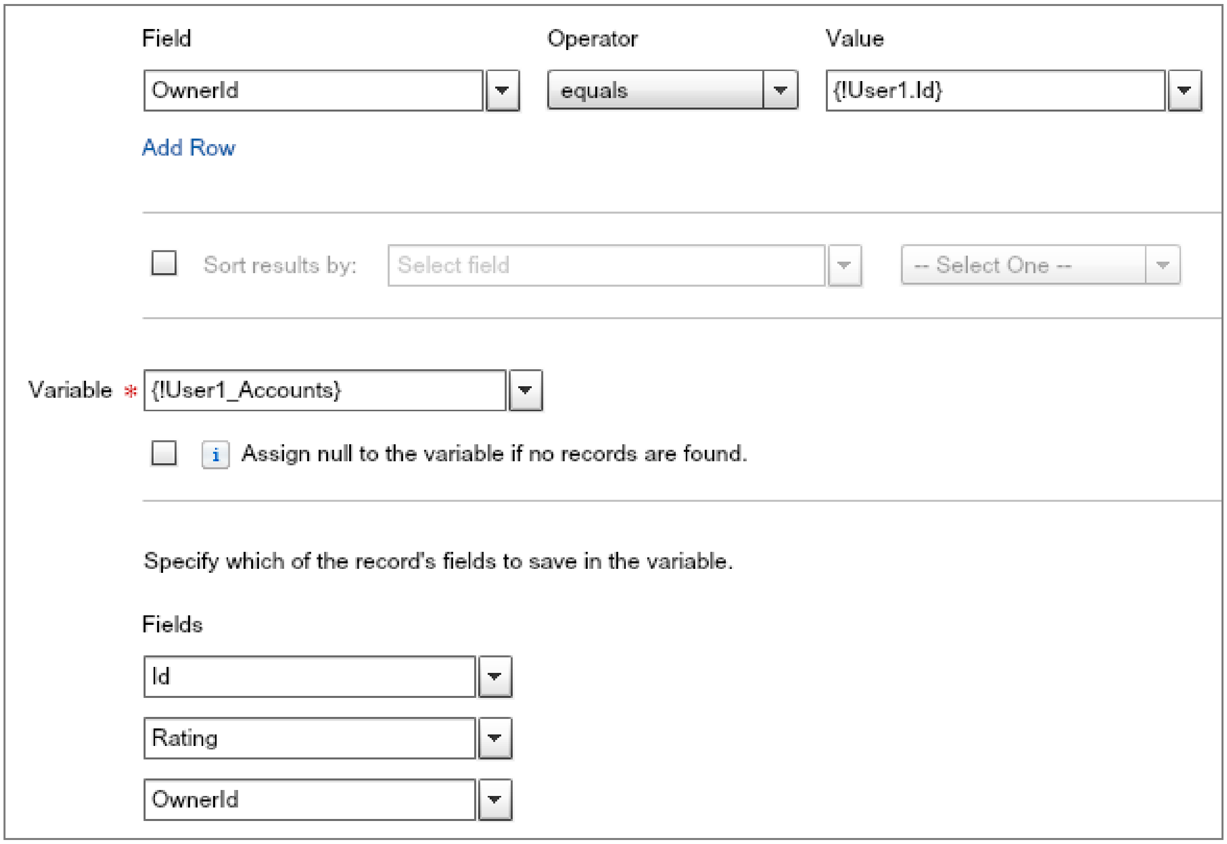
Field	Value
`Field`	*`OwnerId`*

Field	Value
`Operator`	*`equals`*
`Value`	*`{!User1.Id}`*

4. Create an sObject collection variable to store the information that the flow will need to reassign the accounts,.
   a. For the `Variable` field, click ▾ and select **CREATE** > **NEW SObject Collection Variable**.
   b. Enter the following values.

Field	Value
`Unique Name`	*`User1_Accounts`*
`Description`	*`Stores the account’s ID, rating (hot, warm, or cold), and owner ID`*
`Input/Output Type`	*`Input and Output`*
`Object Type`	*`Account`*

   c. Click **OK**.
5. Store the `Id`, `Rating`, `OwnerId` field values in the sObject collection variable.
   a. Click **Add Row** twice.
   b. For `Fields`, type *`Id`*, and select **Id** from the picklist.
   c. In the next row, type *`Rating`*, then select **Rating** from the picklist.
   d. In the last row, type *`OwnerId`*, then select **OwnerId** from the picklist.
6. Verify that your overlay looks like this, and then click **OK**.



Field	Operator	Value
OwnerId	equals	{!User1.Id}

□ Sort results by: Select field -- Select One --

Variable * {!User1_Accounts}

□ Assign null to the variable if no records are found.

Specify which of the record's fields to save in the variable.

Fields

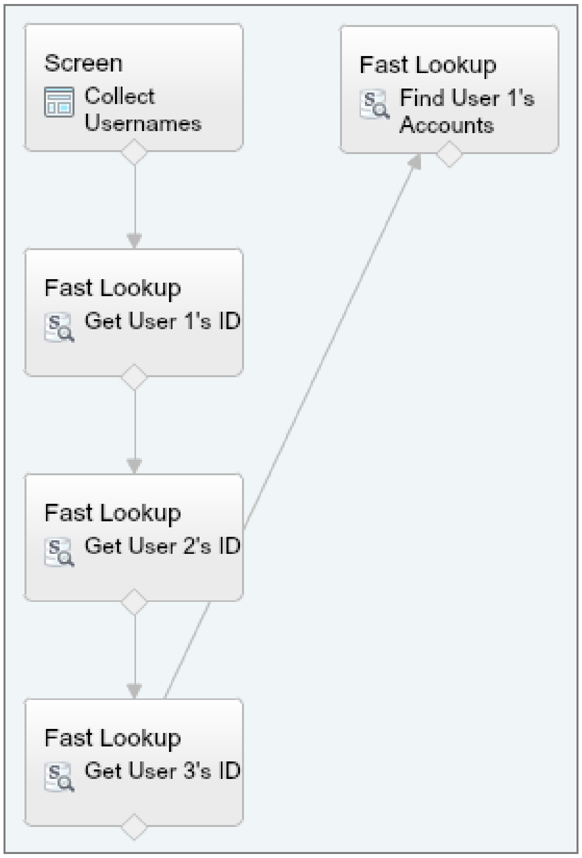
Id

Rating

OwnerId

7. Drag the node from the bottom of Get User 3's ID onto Find User 1's Accounts to connect the elements.
8. Just to be safe, save your progress.
   a. On the button bar, click **Save**.
   b. Ignore any warnings that appear, and click **OK**.

Your flow should now look something like this.



Step 5: Reassign Accounts

The flow reassigns account ownership in two steps. First, it evaluates the accounts one at a time according to rating. Then, depending on whether the account was hot, warm, or cold, the flow assigns it to a new owner.

In the following steps, you'll use a Loop element to iterate through each account and then Decision and Assignment elements to assign it to the right employee.

If you've worked with loops before, you know that they're useful when you need to perform the same task over and over again—especially if you don't know how many times you'll need to repeat that task. In this case, you want to change owners for a set of account records based on the `Rating` field value. But a Loop element can only process one item at a time. Since sObject collection variables store multiple items, you need a *loop variable* to bridge the gap. A loop variable is simply an sObject variable that the loop uses while iterating. When a loop starts, the first item in the collection variable is copied into the loop variable. Any elements inside the loop then act on the loop variable. Once the iteration is over, a new item is loaded into the loop variable, which repeats until there are no items left in the collection to process.

So, when working with Loop elements in the Cloud Flow Designer, you need to keep two things in mind. First, if you want to update an item's field values inside of a loop, you need to make those updates to the loop variable. Second, once a loop variable is acted upon by elements in the loop, any updated field values must be copied to another collection variable, or else they are lost the next time the loop iterates.

IN THIS SECTION:

Loop Through the Account Records with a Loop Element

In this step, the flow uses a Loop element to iterate through the items in the sObject collection variable. Because a Loop element can only process one item at a time, you'll also create a loop variable in which to store individual items from the collection variable as the loop iterates.

Determine Who Gets New Accounts with a Decision Element

In this step, the flow uses a Decision element to evaluate the Rating field of each item when it's stored in the loop variable. If the item has a Hot rating, the flow is routed to an element that assigns ownership of the item to User 2. If the item has a Warm or Cold rating, the flow is routed to an element that assigns ownership of the item to User 3.

Reassign Account Ownership with Assignment Elements

In this step, the flow assigns the account whose values are stored in the loop variable to a new owner. If the Hot outcome occurs, the account is assigned to User 2. If the default outcome occurs, the account is assigned to User 3. You'll do this by using a pair of Assignment elements.

Copy Values from the Loop Variable to a New Collection Variable

In the previous steps, the flow looked at the Rating field of the item and assigned it to User 2 or User 3 depending on whether it was hot, warm, or cold. These updates are stored in the loop variable, however, so, in this step, you'll create a new sObject collection variable to make sure the they don't get lost the next time the loop iterates.

Loop Through the Account Records with a Loop Element

In this step, the flow uses a Loop element to iterate through the items in the sObject collection variable. Because a Loop element can only process one item at a time, you'll also create a loop variable in which to store individual items from the collection variable as the loop iterates.

1. From the Palette tab in the left pane, click and drag a Loop element onto the canvas.
2. For Name, type *Reassign Accounts*.
   The Unique Name field is automatically populated based on this entry.
3. Change Unique Name to *Reassign_User1s_Accounts*.
4. For Loop through, type *User1_Accounts*, and select **User1_Accounts** from the picklist.

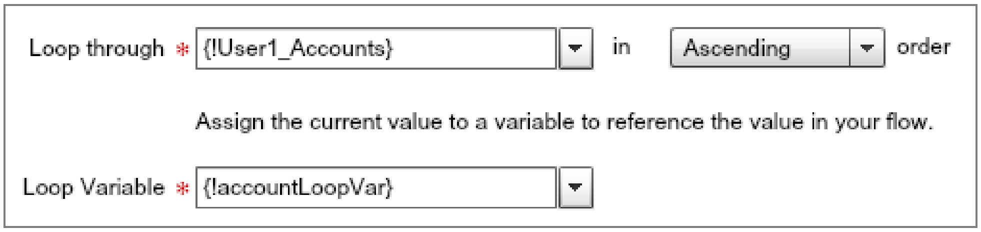
   This tells the loop which sObject collection variable to step through when assigning a new item to the loop variable.

   For order, select **Ascending**.
5. For Loop Variable, click ▼ and select **CREATE NEW** > **SObject Variable**.
   a. Configure the variable.

Field	Value
Unique Name	*accountLoopVar*
Description	*Stores an account record during one loop iteration*
Input/Output Type	*Input and Output*
Object Type	*Account*

   b. Click **OK**.

6. Verify that your overlay looks like this, and then click **OK**.



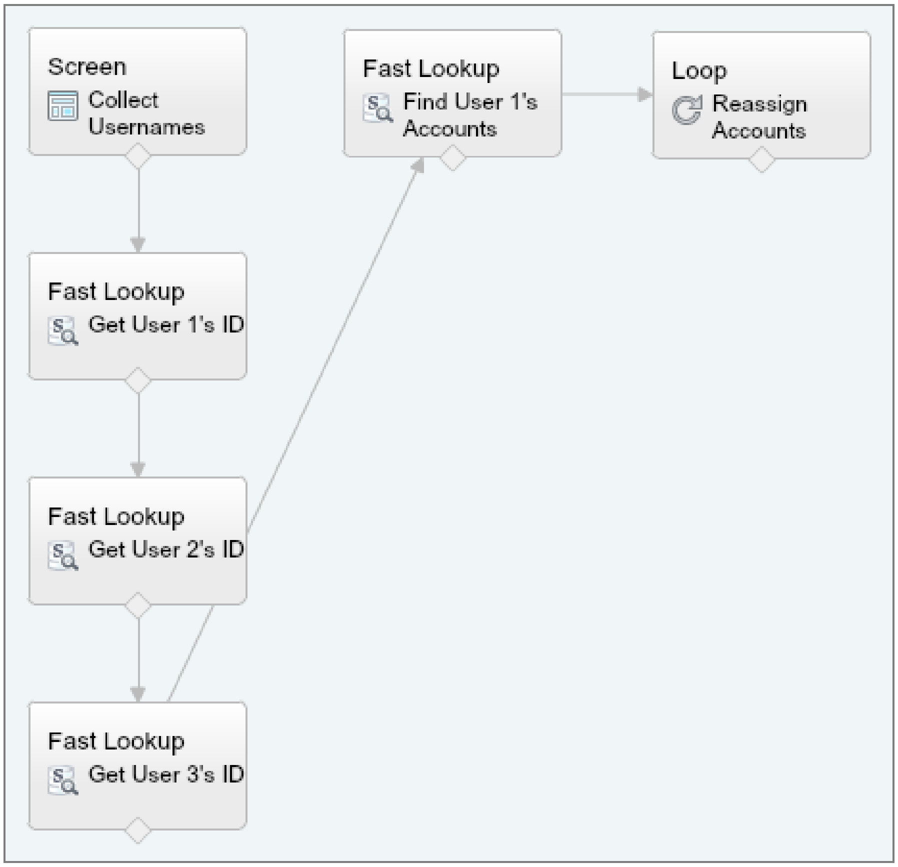
Loop through * {!User1_Accounts} in Ascending order

Assign the current value to a variable to reference the value in your flow.

Loop Variable * {!accountLoopVar}

7. Drag the node from the bottom of Find User 1's Accounts onto Reassign Accounts to connect the elements.

The flow should now look something like this.



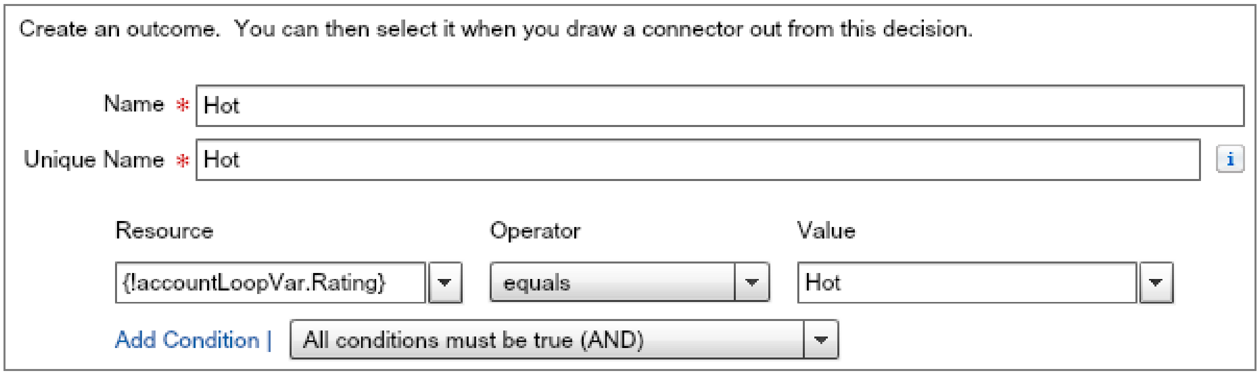
Determine Who Gets New Accounts with a Decision Element

In this step, the flow uses a Decision element to evaluate the `Rating` field of each item when it's stored in the loop variable. If the item has a `Hot` rating, the flow is routed to an element that assigns ownership of the item to User 2. If the item has a `Warm` or `Cold` rating, the flow is routed to an element that assigns ownership of the item to User 3.

1. From the Palette tab in the left pane, click and drag a Decision element onto the canvas.
2. For `Name`, type *`What's the Account Rating?`*.
   The `Unique Name` field is automatically populated based on this entry.

3. Create an outcome for hot accounts.
   a. In the Outcomes section, for `Name`, type **Hot**.
   The `Unique Name` field is automatically populated based on this entry.
   b. Set the outcome criteria.

Field	Value
`Resource`	*`{!accountLoopVar.Rating}`*
`Operator`	*`equals`*
`Value`	*`Hot`*



Create an outcome. You can then select it when you draw a connector out from this decision.

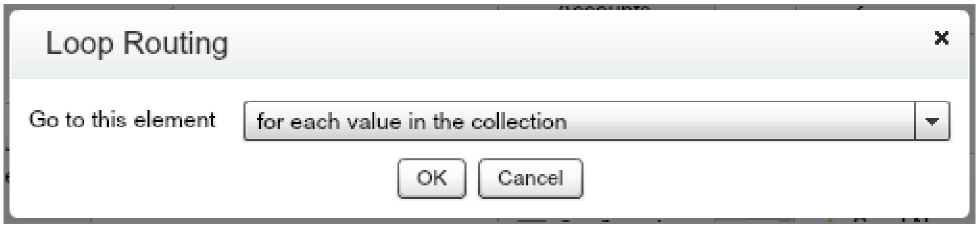
Name * Hot

Unique Name * Hot

Resource	Operator	Value
{!accountLoopVar.Rating}	equals	Hot

Add Condition | All conditions must be true (AND)

4. Configure the default outcome, which occurs when accounts don't meet the criteria for the `Hot` outcome—in other words, the account is `Warm`, `Cold`, or doesn't have a rating set.
   a. Click **[Default Outcome]**.
   b. For `Name`, replace the predefined value with *`Warm or Cold`*.
5. Click **OK**.
6. Drag the node from the bottom of Reassign Accounts onto What's the Account Rating to connect them.
   The Loop Routing overlay appears.
7. Verify that `Go to this element` is set to **for each value in the collection**.
   This selection tells the flow to execute the Decision element every time a new item's field values are copied into the loop variable. This makes sense, since the flow needs to look at the item's rating before deciding whether to assign it to User 2 or User 3.

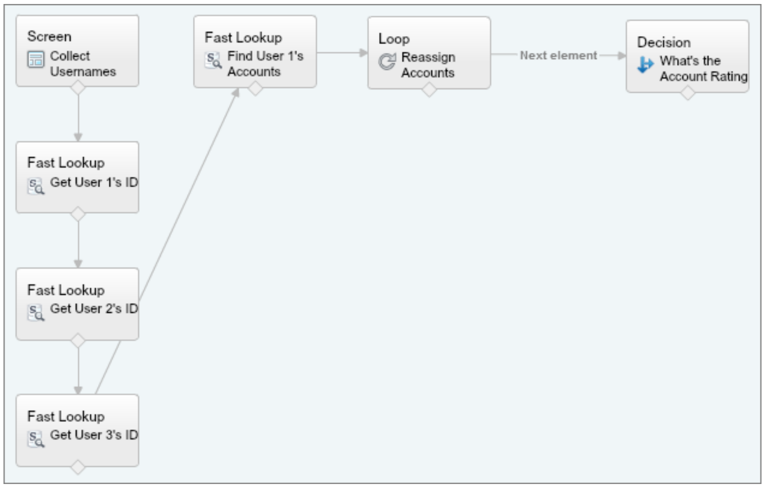


Loop Routing

Go to this element: for each value in the collection

8. Click **OK**.

The flow should now look something like this.



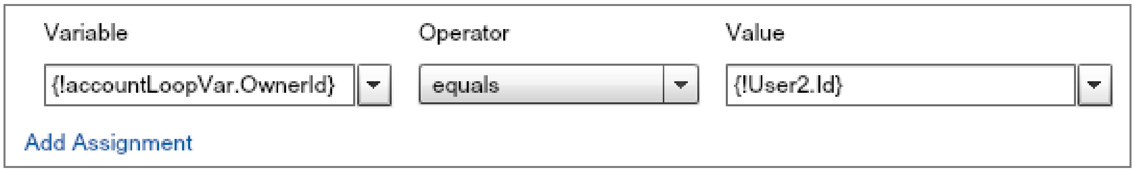
Reassign Account Ownership with Assignment Elements

In this step, the flow assigns the account whose values are stored in the loop variable to a new owner. If the `Hot` outcome occurs, the account is assigned to User 2. If the default outcome occurs, the account is assigned to User 3. You'll do this by using a pair of Assignment elements.

1. Assign hot accounts to User 2.
   - a. From the Palette tab in the left pane, click and drag an Assignment element onto the canvas.
   - b. For `Name`, type *`Assign to User 2`*.
     The `Unique Name` field is automatically populated based on this entry.
   - c. Update the `OwnerId` field value in the loop variable to match User 2's ID.

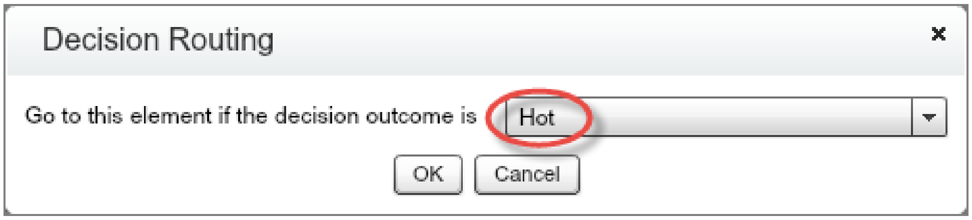
Field	Value
`Variable`	*`{!accountLoopVar.OwnerId}`*
`Operator`	*`equals`*
`Value`	*`{!User2.Id}`*

   - d. Verify that your screen overlay looks like this, and then click **OK**.



Variable	Operator	Value
{!accountLoopVar.OwnerId}	equals	{!User2.Id}

2. Drag the node from the bottom of What's the Account Rating onto Assign to User 2 to connect the elements.
3. Verify that the decision routing dialog box is set to **Hot**.



Decision Routing

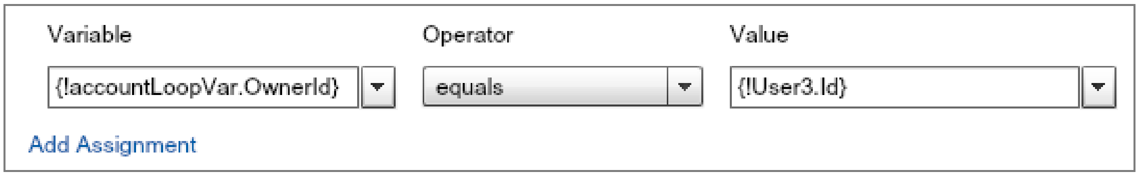
Go to this element if the decision outcome is: Hot

Now, when the item in the loop variable has a `Hot` rating, the flow is routed to the right spot: an Assignment element that transfers the account to User 2.

4. Click **OK**.
5. Assign warm and cold accounts to User 3.
   a. From the Palette tab in the left pane, click and drag an Assignment element onto the canvas.
   b. For `Name`, type *`Assign to User 3`*.
      The `Unique Name` field is automatically populated based on this entry.
   c. Update the `OwnerId` field in the loop variable to match User 3's ID.

Field	Value
`Variable`	*`{!accountLoopVar.OwnerId}`*
`Operator`	*`equals`*
`Value`	*`{!User3.Id}`*

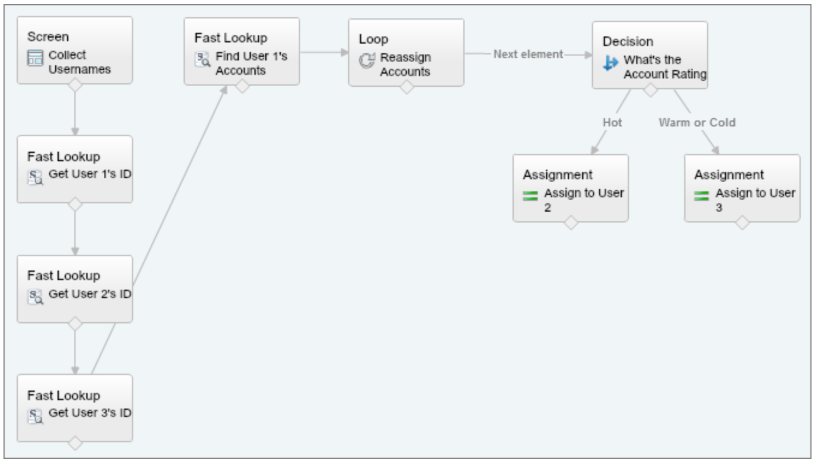
   d. Verify that your screen overlay looks like this, and then click **OK**.



Variable	Operator	Value
{!accountLoopVar.OwnerId}	equals	{!User3.Id}

6. Drag the node from the bottom of What's the Account Rating onto Assign to User 3 to connect the elements.
   Since you've already set a path for the `Hot` outcome, the flow automatically assigns the default outcome to this path.

The flow should now look something like this.



Copy Values from the Loop Variable to a New Collection Variable

In the previous steps, the flow looked at the `Rating` field of the item and assigned it to User 2 or User 3 depending on whether it was hot, warm, or cold. These updates are stored in the loop variable, however, so, in this step, you'll create a new sObject collection variable to make sure the they don't get lost the next time the loop iterates.

Note: You may be asking yourself why you need to add another sObject collection variable. After all, don't you already have one? Can't you just update it? You're correct: there is another sObject collection variable in the flow—it's even set up to store accounts—but we can't use it here. In the Cloud Flow Designer, you can add or clear items in a collection variable, but not update them.

1. From the Palette tab in the left pane, click and drag an Assignment element onto the canvas.
2. For `Name`, type *`Add Values to Updated Collection`*.
   The `Unique Name` field is automatically populated based on this entry.
3. Create an sObject collection variable to store the updated assignments.
   a. For the `Variable` field, click ▼ and select **CREATE** > **NEW SObject Collection Variable**.
   b. Enter the following values.

Field	Value
`Unique Name`	*`coll_updatedAccts`*
`Description`	*`Stores updated account records as the loop iterates`*
`Input/Output Type`	*`Input and Output`*
`Object Type`	*`Account`*

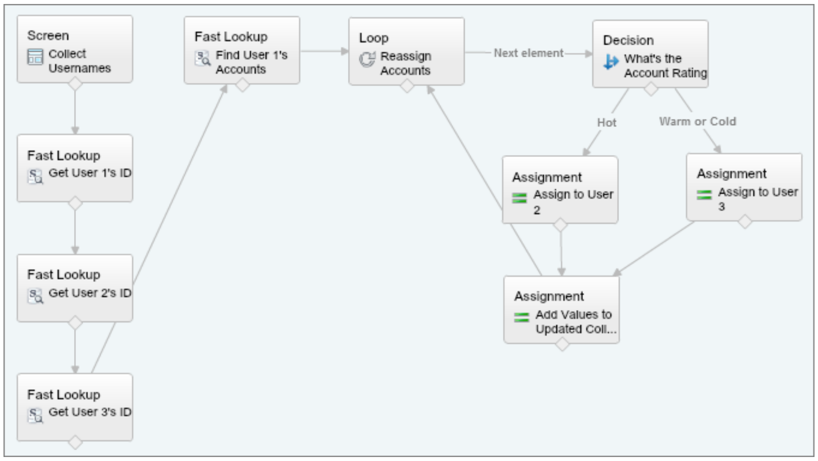
   c. Click **OK**.
4. Configure the other fields so that the current item in the loop variable gets added to `{!coll_updatedAccts}`.

Field	Value
`Operator`	*add*
`Value`	*{!accountLoopVar}*

5. Connect the Assignment elements so that no matter which user is assigned the account, the sObject collection variable gets updated.
   a. Drag the node from the bottom of Assign to User 2 onto Add Values to Updated Collection.
   b. Drag the node from the bottom of Assign to User 3 onto Add Values to Updated Collection.
6. Close the loop by connecting Add Values to Updated Collection to Reassign Accounts.

   By closing the loop, the flow can move on to the next item in the collection after `{!coll_updatedAccts}` gets updated.
7. Just to be safe, save your progress.
   a. On the button bar, click **Save**.
   b. Ignore any warnings that appear, and click **OK**.

The flow should now look something like this.



Step 6: Update the Account Records

Once the flow assigns the accounts to new owners, it needs to update Salesforce so that the changes are visible to other employees at Acme Wireless.

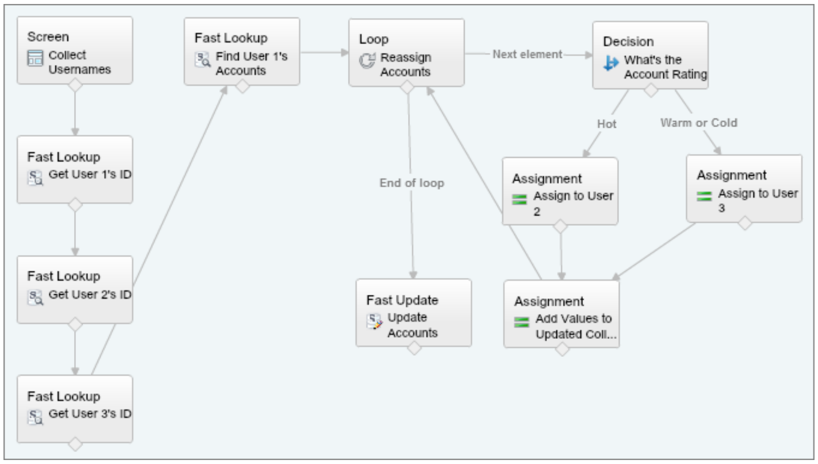
In this step, the flow will update the accounts using the values stored in the sObject collection variable `{!coll_updatedAccts}`. To do this, you'll use a Fast Update element, which can update all of the records in one step. In a Fast Update element, the flow uses an sObject variable or sObject collection variable to update one or more records in Salesforce. Unlike a Record Update element, Fast Updates don't require much configuration. All you need to do is specify which flow variable contains the updated values, and double-check that the variable includes the record's ID, which is how the flow finds the right records to update. In this case, the flow will look up account records using the `Id` field values stored in the sObject variable, and then copy the updated `OwnerId` field values to the corresponding fields in the records.

Note: Earlier in the tutorial, you learned that you could have used a Record Lookup element in place of a Fast Lookup, although it would have meant a bit more work since you'd have to create extra variables. In this step, you wouldn't be able to substitute a Record Update element for the Fast Update. Why? The Record Update element doesn't accept sObject collection variables, which is where the account updates are being stored.

1. From the Palette tab in the left pane, click and drag a Fast Update element onto the canvas.
2. For `Name`, type *`Update Accounts`*.
   The `Unique Name` field is automatically populated based on this entry.
3. For `Variable`, type *`{!coll_updatedAccts}`*, and select **{!coll_updatedAccts}** from the picklist.
4. Click **OK**.
5. Drag the node from the bottom of Reassign Accounts onto Update Accounts to connect the elements.

   This is the path the flow will execute after the loop has processed all of the items in the sObject collection variable. This makes sense, since we don't want the flow to update the account records until the loop has finished assigning a new owner to every item in the collection.

Your flow should now look something like this.



Step 7: Confirm the Updates

Until this point, the flow has worked mostly in the background to reassign accounts. So, now's a good time to check in and let the manager know that everything went off without a hitch.

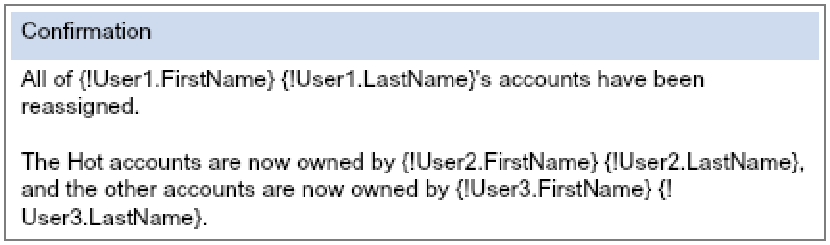
In this step, you'll add a Screen element confirming that the flow successfully reassigned User 1's records to Users 2 and 3.

1. From the Palette tab in the left pane, click and drag a Screen element onto the canvas.
2. For `Name`, type *`Confirmation`*.
   The `Unique Name` field is automatically populated based on this entry.
3. Confirm which employees had their accounts reassigned.
   a. From the Add a Field tab, double-click **Display Text**. A blank display text field appears in the preview pane.
   b. Click **[Display Text]** in the preview pane.
   c. Configure the field.

Field	Value
`Unique Name`	*`ReassignMessage`*

text box	*`All of {!User1.FirstName} {!User1.LastName}'s accounts have been reassigned.`*
	`The Hot accounts are now owned by {!User2.FirstName} {!User2.LastName}, and the other accounts are now owned by {!User3.FirstName} {!User3.LastName}.`

4. Verify that your preview pane looks like this, and then click **OK**.



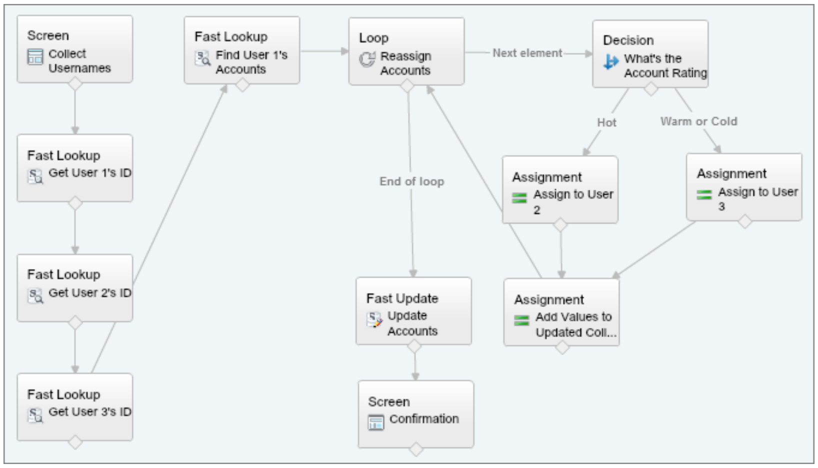
Confirmation

All of {!User1.FirstName} {!User1.LastName}'s accounts have been reassigned.

The Hot accounts are now owned by {!User2.FirstName} {!User2.LastName}, and the other accounts are now owned by {!User3.FirstName} {!
User3.LastName}.

5. Drag the node from the bottom of Update Accounts onto Confirmation to connect the elements.

Your flow should now look something like this.



Step 8: Email the New Account Owners

The employees who now own the hot, warm, and cold accounts will want to know that they have new customers to contact, so the flow should contact them to share the news.

For this, you'll use a Send Email element, which allows you to quickly create emails to send to Salesforce users.

1. From the Palette tab in the left pane, click and drag a Send Email element onto the canvas.

2. For `Name`, type *`Send New Account Notification`*.
   The `Unique Name` field is automatically populated based on this entry.
3. Create a text template for the message body.

   Tip: You could simply enter the text of the email into the field, but text templates allow you to apply basic formatting to your messages, making them easier for your users to read.

   a. For `Body`, click ▾ and select **CREATE NEW** > **Text Template**.
   b. Configure the text template.

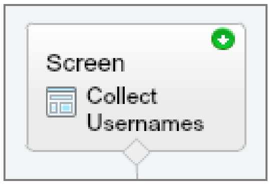
Field	Value
`Unique Name`	*`NewAccountEmail`*
`Description`	*`Email notification to the new owners of hot, warm, and cold accounts`*

text box	*`Hello {!User2.FirstName} and {!User3.FirstName},`*
	`Your account assignments were recently updated. Please log in to Salesforce to check your new accounts. Thank you!`

   c. Click **OK** to save the text template.
4. For `Subject`, type *`New Account Assignments`*
5. Enter the email addresses of the employees getting the new accounts.
   a. Click `Add Row` to add a field for the email addresses.

      The Send Email element lets you add email addresses in two ways: manually, with each address separated by commas, or as part of a collection. If you're only emailing a few users, addresses separated by commas is quickest. If you're sending emails to a large number of users, consider using a collection variable of type Text, since entering many addresses by hand is tedious and prone to error.

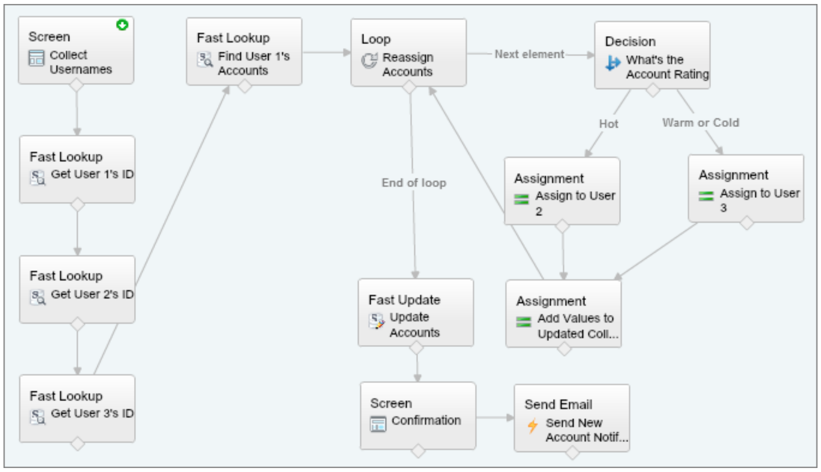
   b. Click ▾ and select **Email Addresses (comma-separated)**.
   c. For `Email Addresses (comma-separated)`, type *`{!User2.Email}, {!User3.Email}`*.
6. Click **OK** to save the Send Email element.
7. Drag the node from the bottom of Confirmation on Send New Account Notification to connect the elements.
8. Designate where the flow should start.
   a. Hover your mouse over Collect Usernames, and three icons appear in the upper right corner.



b. Click ⊕ to set Collect Usernames as the start element.

9. On the button bar, click **Save**.

Your finished flow should look something like this.

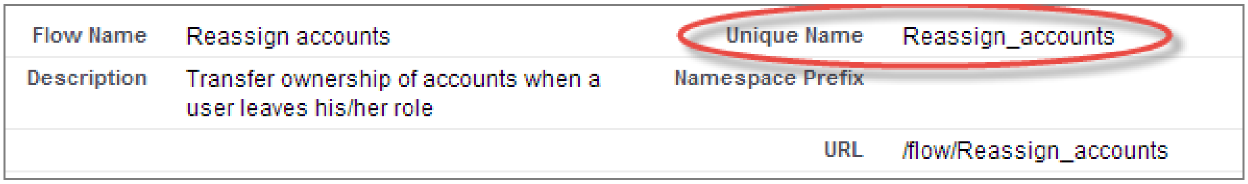


Step 9: Implement the Flow by Using a Visualforce Tab

This flow makes it a lot simpler for managers to transfer accounts between employees, but you still need to give them an easy way to access it.

You'll implement this flow by using a Visualforce tab, which combines a Visualforce page and a Visualforce tab. The Visualforce page runs the flow and allows you to designate which users in your organization can access it. The tab is how users will navigate to the flow in their instance of Salesforce.

1. Get the unique name of the flow.
   a. On the button bar, click **Close**.
   The Cloud Flow Designer closes and the flow detail page opens.
   b. Copy the flow's unique name.



Flow Name	Reassign accounts	Unique Name	Reassign_accounts
Description	Transfer ownership of accounts when a user leaves his/her role	Namespace Prefix	
		URL	/flow/Reassign_accounts

2. Create the Visualforce page.
   a. From Setup, enter *`Visualforce Pages`* in the `Quick Find` box, then select **Visualforce Pages**.
   b. Click **New**.
   c. Set the Visualforce page's details.

Field	Value
`Label`	*`Reassign Account Ownership`*
`Name`	*`Reassign_Account_Ownership`*
`Description`	*`Transfer account ownership when a user leaves his/her role`*

d. On the Visualforce Markup tab, delete the default content between the `<apex:page>` tags.

e. Type `<flow:interview name="MyUniqueFlowName"/>` between the tags.

The `<flow:interview>` component embeds the flow within the Visualforce page.

f. For the `Name` attribute, delete *`MyUniqueFlowName`* and paste in the flow name you copied earlier. When you're done, the code should look like this.

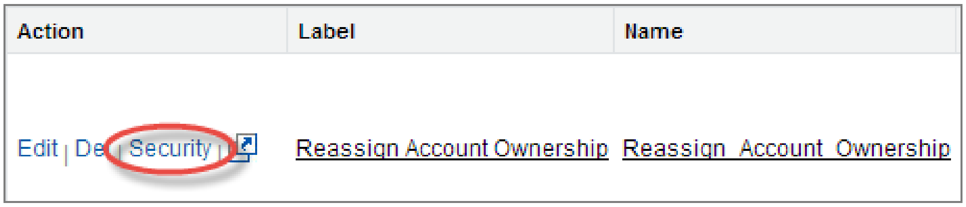
```
<apex:page>
<flow:interview name="Reassign_accounts"/>
</apex:page>
```

g. Click **Save**.

3. Set the Visualforce page's security so that only administrators can access the flow.

a. From Setup, enter *`Visualforce Pages`* in the `Quick Find` box, then select **Visualforce Pages**.

b. In the Action column next to Reassign Account Ownership, click **Security**.



Action	Label	Name
Edit | De | Security |	Reassign Account Ownership	Reassign_Account_Ownership

c. For `Enabled Profiles`, verify that **System Administrator** is the only profile listed.

d. Click **Save**.

4. Create a Visualforce tab so that you can display the Visualforce page to Acme Wireless managers.

a. From Setup, enter *`Tabs`* in the `Quick Find` box, then select **Tabs**.

b. In the Visualforce Tabs related list, click **New**.

c. Configure the Visualforce tab.

Field	Value
`Visualforce Page`	*`Reassign Account Ownership [Reassign_Account_Ownership]`*
`Tab Label`	*`Reassign Accounts`* The `Tab Name` field is automatically populated based on this entry.

Field	Value
`Tab Style`	*`Computer`* Tab Style specifies the color scheme and icon of the Visualforce tab.
`Description`	*`Transfer account ownership when a user leaves his/her role`*

d. Click **Next**.

e. Click **Next** again.

f. Click **Save**.

The tab now appears to the right of your other tabs.

Note: If you walked through the tutorial for helping Acme Wireless measure customer satisfaction, you may have noticed that the Visualforce tab looks a lot like a web tab. You're correct! In fact, you could have used a web tab, but we wanted to show you a different way of doing things in this tutorial.

Step 10: Test the Flow

Testing is an important part of building a flow, allowing you to fine-tune the flow's behavior, identify and fix bugs, and otherwise make sure your users have a pleasant experience. And of course, you benefit, since you're much less likely to spend time later responding to support requests and fixing errors.

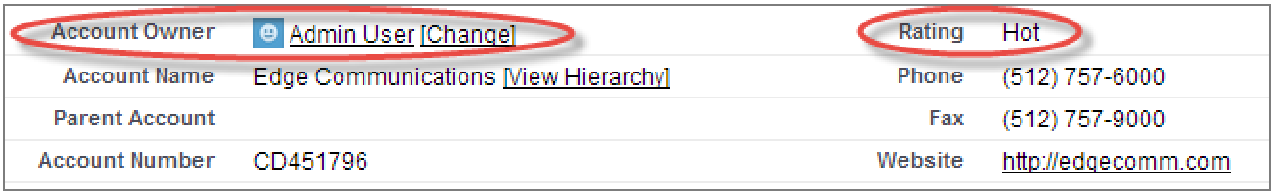
In this step, you'll play the part of an Acme Wireless manager and use the flow to assign new accounts to two sales associates on your team, Renee Moreau and Madison Rigsby. As administrator of the developer organization that you created for this workbook, you own the accounts that will be reassigned, so, along with playing manager, you'll also play the part of the employee who's moving on to the new role.

1. Confirm that you own one of the accounts that you'll be reassigning.

   a. In the global search bar, search for *`Edge Communications`*.

   b. In the search results, click **Edge Communications**.

   c. Verify that you are listed as the owner and the account's rating is `Hot`.



Account Owner	Admin User [Change]	Rating	Hot
Account Name	Edge Communications [View Hierarchy]	Phone	(512) 757-6000
Parent Account		Fax	(512) 757-9000
Account Number	CD451796	Website	http://edgecomm.com

2. Create a user record for Renee, who will be taking over the hot accounts.

   a. In a new browser tab or window, open another instance of your Salesforce organization.

   b. From Setup, enter *`Users`* in the `Quick Find` box, then select **Users**.

   c. Click **New User**.

   d. Configure the record's fields.

Field	Value
`First Name`	*Renee*
`Last Name`	*Moreau* `Alias` is automatically populated based on this entry.
`Email`	Type your email address, since you want to make sure that the flow is sending the confirmation email. `Username` and `Nickname` are automatically populated based on this entry.
`User License`	*Salesforce*
`Profile`	*Standard User*

As you just saw, Salesforce uses an email address to generate values for the `Username` and `Nickname` fields. These field values need to be unique, or else Salesforce won't let you save the user record. Because you'll be creating a second user record using your email address, you need to add some text to the username and nickname now to make them unique.

e. For `Username`, append *_test_1* after your name.

So, if the username was *a.user@salesforce.com*, it would look like *a.user_test_1@salesforce.com*.

f. For `Nickname`, append *_test_1* to the end of the nickname.

g. Click **Save**.

Remember to keep this tab open—you copy and paste Renee's username in a few steps.

3. Create a user record for Madison, who will be taking over the warm and cold accounts.

a. In a new browser tab or window, open another instance of your Salesforce organization.

b. From Setup, enter *Users* in the `Quick Find` box, then select **Users**.

c. Click **New User**.

d. Configure the record's fields.

Field	Value
`First Name`	*Madison*
`Last Name`	*Rigsby* `Alias` is automatically populated based on this entry.
`Email`	Type your email address, since you want to make sure that the flow is sending the confirmation email. `Username` and `Nickname` are automatically populated based on this entry.
`User License`	*Salesforce*
`Profile`	*Standard User*

Just like before, you need to add some text to the username and nickname to make them unique.

e. For `Username`, append *`_test_2`* after your name.

So, if the username was *`a.user@salesforce.com`*, it would look like *`a.user_test_2@salesforce.com`*.

f. For `Nickname`, append *`_test_2`* to the end of the nickname.

g. Click **Save**.

Remember to keep this tab open—you'll be copying and pasting Madison's username in the next step.

4. Reassign the accounts.

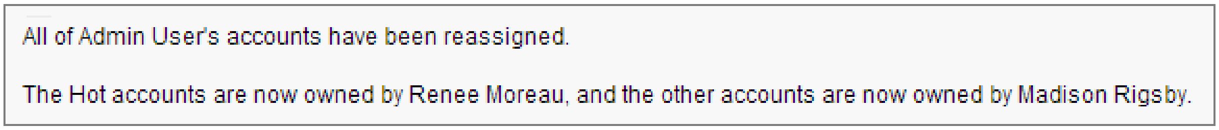
a. Click the Reassign Accounts tab.

b. For `User whose accounts need to be reassigned`, type your username into the field.

c. For `Assign hot accounts to`, paste Renee's username into the field.

d. For `Assign other accounts to`, paste Madison's username into the field.

e. Click **Next**.
The flow displays a screen to confirm that the accounts have been successfully reassigned.



All of Admin User's accounts have been reassigned.

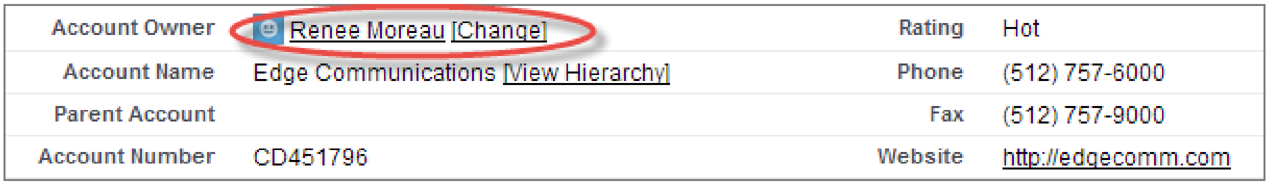
The Hot accounts are now owned by Renee Moreau, and the other accounts are now owned by Madison Rigsby.

5. Confirm that the account you viewed earlier has been successfully reassigned.

a. In the global search bar, search for *`Edge Communications`*.

b. In the search results, click **Edge Communications**.

c. Since the account is `Hot`, verify that Renee is now the owner.



Account Owner	Renee Moreau [Change]	Rating	Hot
Account Name	Edge Communications [View Hierarchy]	Phone	(512) 757-6000
Parent Account		Fax	(512) 757-9000
Account Number	CD451796	Website	http://edgecomm.com

You're done—take a second to pat yourself on the back!

You've accomplished a lot in this tutorial, but there are many ways to expand on what you've built in this flow by using other objects in Salesforce. For example, you could look up all of the contacts that are associated with each account and use that information to schedule follow-up calls or emails to the customers. Since Salesforce can track a customer's preferred contact method, you can also configure the flow to tell the accounts' new owners the best way to get in touch.