



Mel Scott
Certified Salesforce.com Consultant

mel@meloniescott.com

310-925-3914
www.meloniescott.com



Take “Action” with Flows

FORCE ACADEMY LOS ANGELES MAY 13, 2019
Session Training Instructions for:

Using a Flow to create a field update Account Action

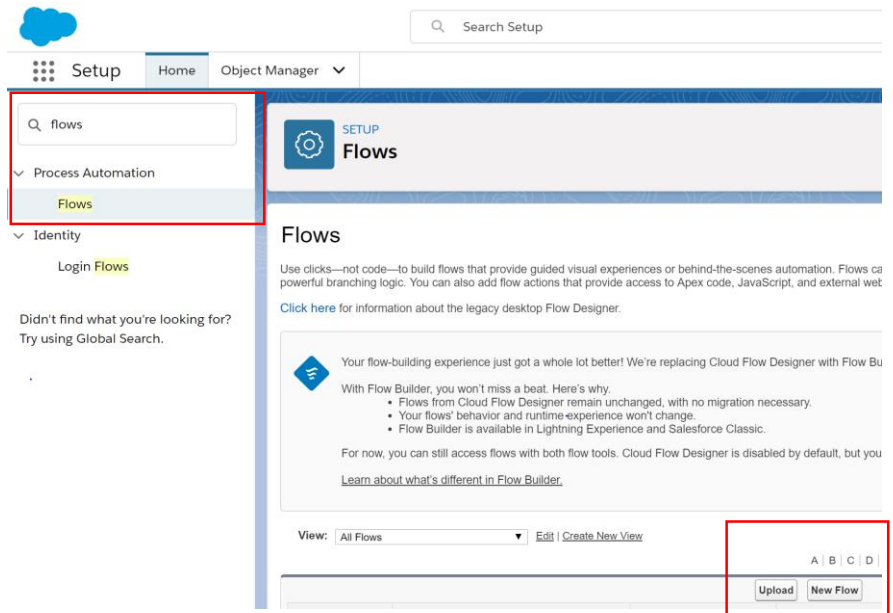
Use Case: *Allow Account owners to update the Contact addresses with the Account’s address at their discretion, not an automatic process.*

Presented by Melonie Scott, Independent Consultant



1

Create a New Flow from Setup



2

Data:
Get Records – *you need to collect the record data you will reference*

Toolbox

Elements Manager

✓ User Interface (1)

Screen

✓ Logic (4)

Assignment

Decision

Pause

Loop

✓ Data (4)

Create Records

Update Records

Get Records

Delete Records



3

- Add Label
- Description
- Select the Object
(Account)

4

- Condition Requirements =
Conditions are Met
- Field is ID (Account ID)
- Only the first record
- In separate variables

5

- Click into Value



Create a New Resource:

you need a destination to house the record data you are referencing

- Resource Type = Variable
- API Name = **recordId**
- Add Description
- Data Type = Text
- Default Value left blank
- Check off both Input and Output

The screenshot shows the 'New Get Records' dialog box in Salesforce. The main title is 'New Get Records' with a close button (X) in the top right corner. Below the title bar, there is a section for 'Get Records of This Object' and a 'New Resource' configuration form. The form includes the following fields and options:

- Resource Type:** A dropdown menu set to 'Variable'.
- API Name:** A text input field containing 'recordId'.
- Description:** A text area containing 'Description here - Account ID'.
- Data Type:** A dropdown menu set to 'Text'. To its right is a checkbox labeled 'Allow multiple values (collection)' which is unchecked.
- Default Value:** A text input field with the placeholder 'Enter value or search resources...' and a search icon (magnifying glass) on the right.
- Availability Outside the Flow:** Two checkboxes, both checked:
 - Available for input
 - Available for output

At the bottom of the dialog, there are two sections with 'Cancel' and 'Done' buttons:

- A light gray bar with 'Cancel' and 'Done' buttons.
- A dark gray bar with 'Where to Store Field Values' text and 'Cancel' and 'Done' buttons.



6b

You are back on the Get New Record page:

- Select the Variable (**recordId**) that you just created

Filter Account Records

Condition Requirements
Conditions are Met

Field: Id Operator: Equals Value: {!recordId}

+ Add Condition

6c

Scroll down

- Sort Account Records = Not Sorted
- How Many Records = Only the first record
- Where to Store =
- Where to Store = In separate variables
(we create one variable for each field we want to update)

New Get Records

Field: Id Operator: Equals Value: {!recordId}

+ Add Condition

Sort Account Records

Sort Order: Not Sorted

⚠ If you store only the first record, filter by a unique field, such as ID.

To use the returned **Account** records in the flow, store their fields in variables.

How Many Records to Store

- Only the first record
- All records

Where to Store Field Values

- Together in a record variable
- In separate variables

When no records are returned, set specified variables to null.

Select Variables to Store Account Fields

Field: Search fields... Variable: Enter value or search resources...

Cancel Done



6d

Now we will add each field we are updating. To do this, we need to create the field Variables that will be updated with the Account reference data for each field. We do this just like we created the Account ID Variable in Step 6a.

- For our demo it will be Shipping Street, Shipping City, Shipping State and Shipping Zip Code

Sort Account Records

Sort Order
Not Sorted ▼ ⚠ If you store only the first record, filter by a unique field, such as ID.

To use the returned **Account** records in the flow, store their fields in variables.

How Many Records to Store

- Only the first record
- All records

Where to Store Field Values

- Together in a record variable
- In separate variables
- When no records are returned, set specified variables to null.

Select Variables to Store Account Fields

Field	Variable
Search fields... 🔍	Enter value or search resources... 🔍 🗑️

Cancel Done

Click and select the Acct field
Repeat for each field

Click and select New Resource
Repeat for each field



7

Toolbox

Elements Manager

✓ User Interface (1)

Screen

✓ Logic (4)

Assignment

Decision

Pause

Loop

✓ Data (4)

Create Records

Update Records

Get Records

Delete Records

8a

Next we define what fields to be updated (*the Contact fields*)

- Add Label
- Description
- How to Find = Specify Conditions
- Select the Object
(Contact)
- Condition Requirements = Conditions are Met
 - Field is the AccountId – same as Get Record in Step 6b.
 - Select the same Variable from Step 6b

Edit Update Records

* Label: UpdateContact

* API Name: UpdateContact

Description:

How to Find Records to Update and Set Their Values

Use the IDs and all field values from a record variable or record collection variable

Specify conditions to identify records, and set fields individually

Update Records of This Object Type

* Object: Contact

Filter Contact Records

Condition Requirements: Conditions are Met

Field	Operator	Value
AccountId	Equals	{!recordId}



8b

Each Account field Variable we created in [Step 6d](#) will be mapped (the update) to a Contact field that we will define with Contact field Variables. Again we need to create a Variable for each field to be updated.

- For our demo it will be Mailing Street, Mailing City, Mailing State and Mailing Zip Code

Set Field Values for the Contact Records

Field	Value
MailingStreet	{!AcctShippingStreet}

+ Add Field

9

Each Data section now gets connected to Start





10

Save the Flow

- Type = Screen Flow
(Actions require Screen type)

Save the flow

* Flow Label: ContactUpdateAddressFlowforDemo

* Flow API Name: ContactUpdateAddressFlowforDemo

Description:

* Type: Screen Flow

Show Advanced

Cancel Save

11

Activate the Flow

- Click on the Flow Name to get to the Flow Versions screen
- Click Activate

Action	Flow Label ↑	Description
Open Edit Del	ContactUpdateAddressFlowforDemo	

Click on the Flow Label link

Action	Flow Label	Version	Description
Open Run Activate	ContactUpdateAddressFlowforDemo	1	

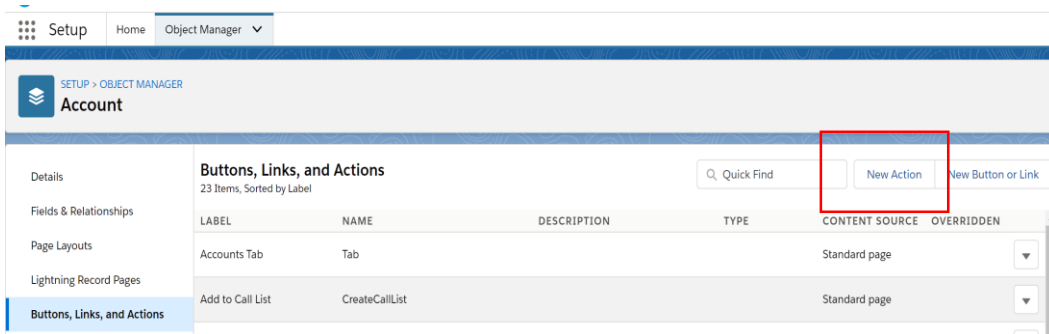
Activate



12

Create the Account Action

- Action Type = Flow
- Flow = Select the Flow we just created
- Standard Label Type = None (we are creating a custom label)
- Add a Label – this will be what the users see on the page
- Add a Description



Account Actions New Action

Enter Action Information

Save Cancel

Object Name Account *i*

Action Type Flow

Flow Screen Flow *i*

Standard Label Type --None-- *i*

Label ContactAddressFlowDemo

Name ContactAddressFlowDemo *i*

Description *i*

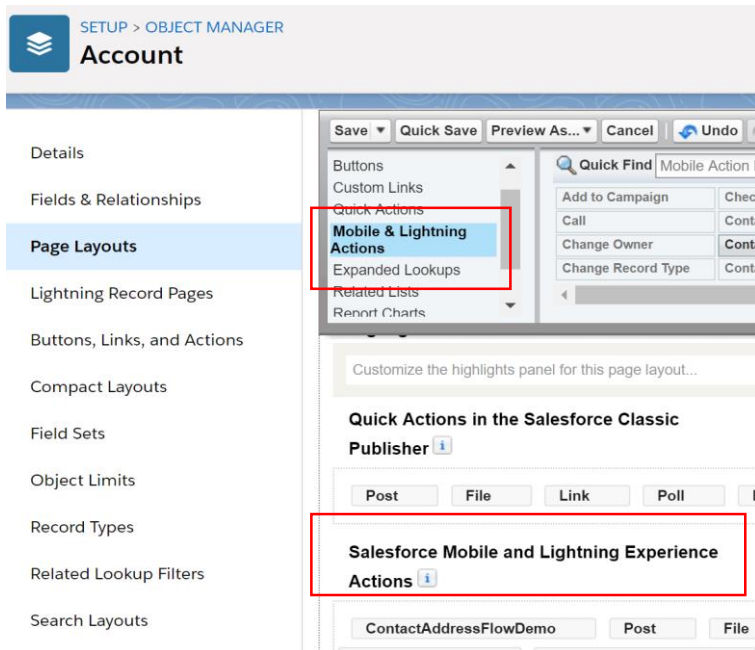
Icon ⚡ [Change Icon](#)

Save Cancel

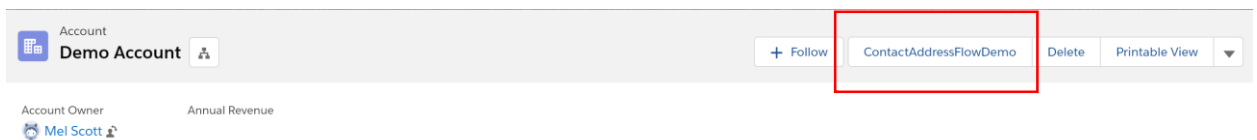


13

Add the Action to the Account Page Layout



The Action now appears on the Account Page Layout



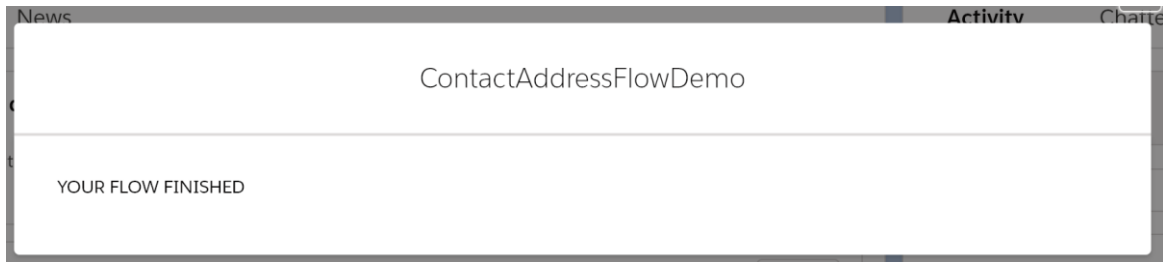


Mel Scott
Certified Salesforce.com Consultant

mel@meloniescott.com

310-925-3914
www.meloniescott.com

Each time you execute the Action a pop up appears confirming the flow has finished.



CONGRATULATIONS!

You have created
an Account Action
using a Flow!

Great Job!!



A few Flow Best Practices



Plan before you start building.

I usually draw out all the details of the business process. Helps me visualize it and I have a clear idea of what information I need, where I'm getting that information from and what logic and actions to perform. Keep things simple.



Now for the Obvious: Build your flows in a sandbox or Developer org.

The last thing you want to do is accidentally change records in production. Build your flows in a sandbox and enter test data. Test various permutations of your flow without worrying about changing or deleting data that your users actually need or use.



So this is Huge!. Save early and often.

You need to know that sometimes the Cloud Flow Designer falls victim to unexpected problems, like losing Internet access. Salesforce does a lot but it just doesn't save your changes automatically. So it's up to you to save your work. Save as often as possible to avoid accidentally losing a few hours' worth of work.



This best practice recommendation is borrowed: Test as many permutations of your flow as you possibly can.

As with all customizations in Salesforce, it's important to test your work. This is especially true if your flow uses branching or other complex logic. Make sure that you test as many possibilities as you can think of before you distribute the flow to your users.

Some thoughts on where you may want to use Flows



Feedback after a manual Event .

You can design a flow after a Case is closed or an Opportunity is won to send an email for feedback
You can use Flow to populate an email template



Action after a system event

Update a team or a customer if a shipment is delayed.
Update a opportunity team if there is news that is relevant.



Transfer record Ownerships

In reorgs or frequently updated territory trees. Transfer ownership of accounts from one user to another by using Object variable collections and loops.