

Build an Integration Procedure with Conditional DataRaptor Loads



NOTE:

Did you sign up for a [special OmniStudio Developer Edition org](#) already? You'll need one to do the steps in this guide. If not, use the link to fill out the form and have an org delivered to your inbox. The Exercise Guide in the first unit of this module has more detailed steps for this process if you need them.

Requirements

The Update Account Primary Contact OmniScript does one of three things: updates the data for the current primary contact, changes the primary contact, or creates a new contact that will be the new primary contact. You must build an Integration Procedure with conditional DataRaptor Loads that can write any of these three changes back into your org.

Prerequisites

- Prepare a JSON with Different Conditional Inputs (prior unit)

Tasks

1. Build an Integration Procedure with Conditional Actions
2. Build a DataRaptor Load to save new primary contact data
3. Build a DataRaptor Load to save updated primary contact data
4. Build a DataRaptor Load to save changed primary contact data

Time

- 30 mins

Task 1: Build an Integration Procedure with Conditional Actions

1. Prepare the Integration Procedure for three new DataRaptors.
 - a. From the **OmniStudio** app dropdown, select **OmniStudio Integration Procedures**.
 - b. Select **team/savePrimaryContactDetails > Team Starter Save Primary Contact Details (Version 1)**.
 - c. Click **Create Version** and edit the name to remove **Starter**. The new name is **Team Save Primary Contact Details**.
 - d. Click **PREVIEW > Edit as JSON**.
 - e. In the **Input Parameters** text box, delete any text that is there, and then paste the JSON you saved when you completed the exercise in the prior unit.



```
{
  "StepContacts": {
    "MsgWeatherAlert": false,
    "RadioPriContact": "Create",
    "BlkUpdatePriContact": {
      "UpdateName": "Leanne
Tomlin",
      "UpdateEmail":
"ltomlin@acme.com",
      "UpdateContactId":
"0036g000002aKzoAAE"
    },
    "BlkChangePriContact": {
      "TChangeContact-Block": 1,
      "TChangeContact":
"Edward Stamos",
      "ChangeContactEmail":
"edward@acme.com",
      "ChangeContactId":
"0036g000002aKzmAAE"
    },
    "BlkCreatePriContact": {
      "CreateFirstName": "aruv",
      "CreateLastName": "anand",
      "CreateEmail":
"aanand@acme.in"
    }
  },
  "AccountId": "0016g000003d0fbAAE"
}
```

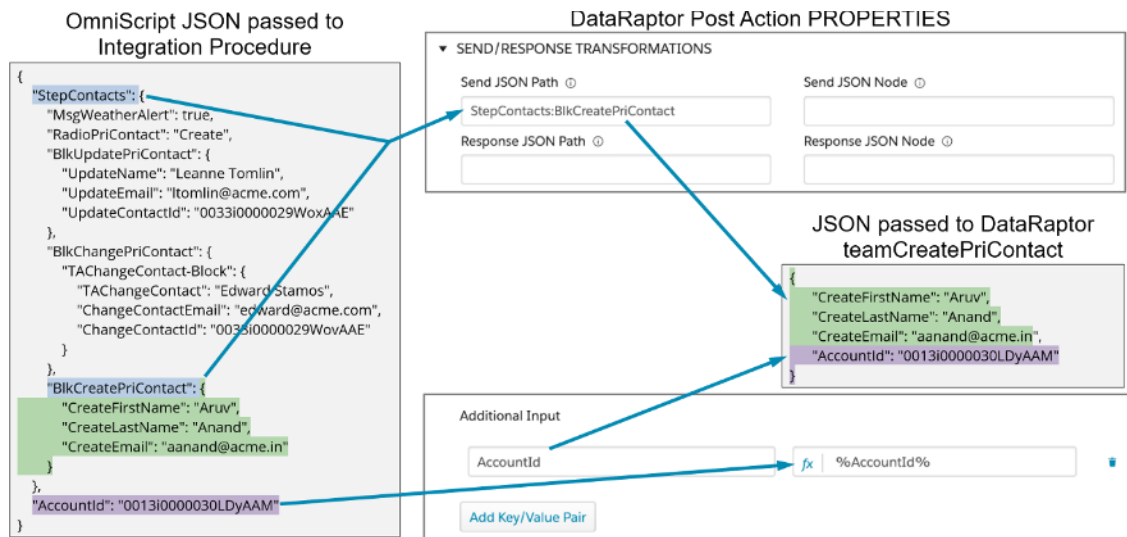
2. Confirm the conditions for when each of the three DataRaptors will execute.

In the **StepContacts** element of your OmniScript, the option selected for **RadioPriContact** defines when a particular DataRaptor will run. You need to confirm that the conditional formula for each DataRaptor in your Integration Procedure matches (Update, Change, or Create).

- a. In the Integration Procedure **Team Save Primary Contact Details**, click **PROPERTIES** and confirm the following:

Element	Field	Value
DRUpdatePriContact	Execution Conditional Formula	StepContacts:RadioPriContact =="Update"
DRChangePriContact	Execution Conditional Formula	StepContacts:RadioPriContact =="Change"
DRCreatePriContact	Execution Conditional Formula	StepContacts:RadioPriContact =="Create"

3. Prepare the data input to use for creating the DataRaptor to create the Primary Contact.



While it's the last of the DataRaptors in the STRUCTURE panel, you will create the Create DataRaptor first because it is the most complex. To create this DataRaptor, you only need a portion of the JSON you pasted into the Preview: the particular node of the array and the AccountId.

- Confirm you are still on the **DRCreatePriContact** element.
- Under **Additional Input**, click **Add Key/Value Pair**.
- In **Additional Input**, enter `AccountId` and in the **fx** field, enter `%AccountId%`. Repeat the step b and c for **DRChangePriContact**.



NOTE:

The value for the key needs to be the AccountId in the input JSON, so you can use a merge field. % is used as the syntax for merge fields in an Integration Procedure and OmniScript.

- Click **PREVIEW**.
- In **Input Parameters** on the left-hand side, change the **RadioPriContact** value to `Create`.

```
    },  
    "BlkCreatePriContact": {  
      "CreateFirstName": "aruv",  
      "CreateLastName": "anand",  
      "CreateEmail": "aanand@acme.in"  
    },  
    "RadioPriContact": "Create"  
  },  
  "AccountId": "0014S000003MwEpQAK"  
}
```

- f. Click **Execute**.
- g. Expand the **Debug Log** and locate **DRCREATEPRICONTACTDEBUG** from the dropdown.
- h. Select and copy the data Input (including the brackets) and save it to re-use below.

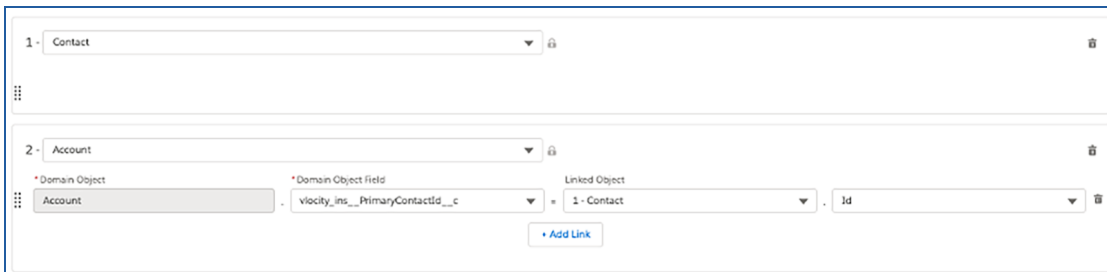
DRCREATEPRICONTACTDEBUG ▼

```
{  
  "Status": true,  
  "Type": "DataRaptor Post Action",  
  "ElapsedTime": 6,  
  "ElapsedTimeCPU": 4,  
  "Input": {  
    "AccountId": "0016g000003d0fbAAE",  
    "CreateEmail": "aanand@acme.in",  
    "CreateLastName": "anand",  
    "CreateFirstName": "aruv"  
  }  
}
```

- i. You are now ready to create the DataRaptor.

Task 2: Build a DataRaptor Load to save new primary contact data

1. Build the new DRCreatePriContact DataRaptor.
 - a. From the dropdown, select **OmniStudio DataRaptor** and open the **teamStarterCreatePriContact** DataRaptor.
 - b. Click **Clone** and name the clone **teamCreatePriContact**, then save.
 - c. The Contact is already mapped, so you are ready to link the Account to the new Contact.
 - d. Click **+ Add Object**, select **Account**, then click **+ Add Link**.
 - e. In the **Domain Object Field**, select **vlocity_cmt__PrimaryContactId__c** or **vlocity_ins__PrimaryContactId__c**.
 - f. In the two **Linked Object** dropdowns, select **1-Contact** and **Id**.



The screenshot shows the DataRaptor configuration interface. It displays two objects: '1- Contact' and '2- Account'. Under the '2- Account' object, the 'Domain Object Field' is set to 'vlocity_ins__PrimaryContactId__c'. The 'Linked Object' is set to '1- Contact' and the 'Id' field is selected. An 'Add Link' button is visible at the bottom.



NOTE:

This is a good example of the power of DataRaptors. You want to use the RecordId of the Contact that was just created to update the value of the Primary Contact field on the Account. The DataRaptor first creates the Contact and then uses data from that newly created record to populate other records linked to it (in this example, the Account).

- g. Click the **FIELDS** sub-tab.
- h. Paste the JSON from the Integration Procedure into the **Input JSON** field.

- i. Confirm you are on the **Account** sub-tab of the **FIELDS** sub-tab and click the **+** to add a new mapping.
- j. Fill in the fields as follows:

Property	Value
Input JSON Path	AccountId
Domain Object Field	Id
Upsert Key	✓
Is Required For Upsert	✓




















Because you need to update the existing Account record, adding an upsert key is required. However, you also want to create a new Contact record.



BEST PRACTICE:

To create a new Contact in a DataRaptor, there are two requirements. Do not create a new Contact when one already exists, so select fields as needed to confirm a Contact does not already exist (in this scenario, first name and last name). The second is to include all the required fields for that type of record in your DataRaptor. (The only field that is required on the Contact record in Salesforce is the Last Name.) Mark any required field as Required for Upsert, so if for some reason there isn't a last name, you won't end up returning an error from Salesforce.

- k. Click the **Contact** sub-tab.
- l. Edit the **CreateLastName** mapping to select **Is Required For Upsert** and **Upsert Key**.
- m. Edit the **CreateFirstName** mapping to select **Upsert Key**.

OBJECTS			FORMULAS	FIELDS	OPTIONS	PREVIEW
1 - Contact	2 - Account	All				
INPUT JSON PATH 			DOMAIN OBJECT FIELD 			
AccountId			1 - Contact	AccountId	 	
CreateEmail			1 - Contact	Email	 	
CreateFirstName			1 - Contact	FirstName	  	
CreateLastName			1 - Contact	LastName	  	
AccountId			2 - Account	Id	  	
			2 - Account	vlocity_ins__PrimaryContactId__c	  	

2. Preview and test the DataRaptor.

- Click **PREVIEW**.
- Paste the JSON from the Integration Procedure into the **Input** text box.
- Click **Execute**.
- There are two records in the **Objects Created** text box: an Account and a Contact.

OBJECTS	FORMULAS	FIELDS	OPTIONS	PREVIEW
Input		Objects Created Performance Metrics - Bro		
<pre>"AccountId": "0013i000006rhJdAAI", "CreateEmail": "aanand@acme.in", "CreateLastName": "Anand", "CreateFirstName": "Aruv"</pre>		<ul style="list-style-type: none"> 1- Account: 0013i000006rhJdAAI 2- Contact: 0033i000003dWR8AAM 		

- Click **Account** and confirm on the Acme details tab that **Aruv Anand** is now the **Primary Contact**, then close the browser tab.
- Click the **Contact** link and confirm that **Aruv Anand** has been created and linked to **Acme** correctly.

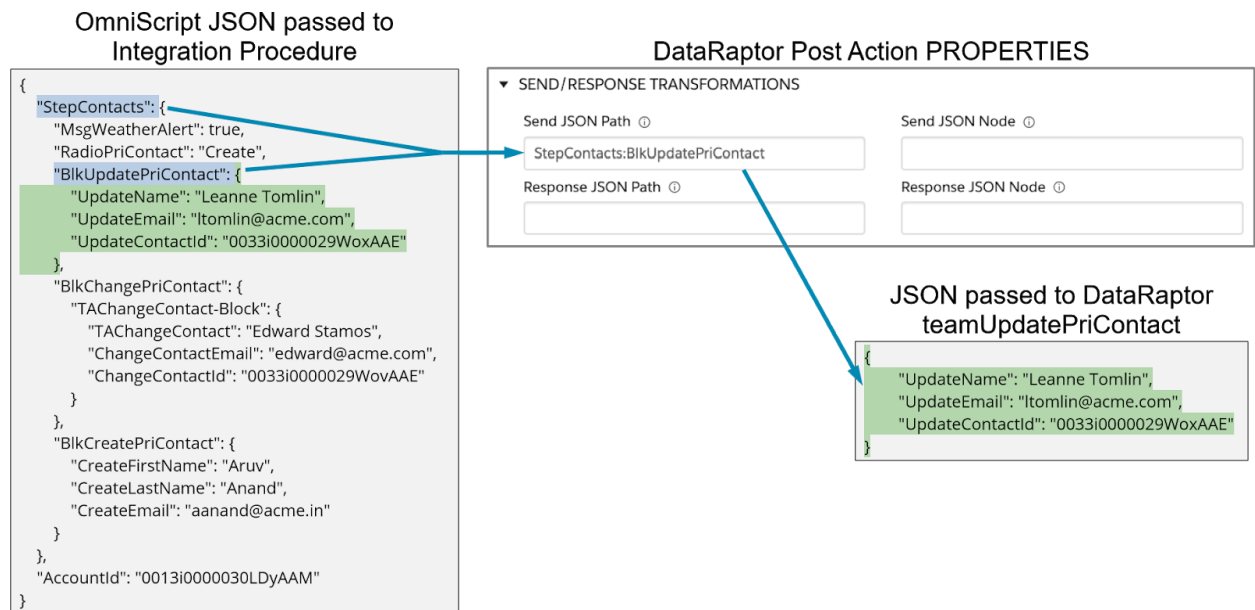
3. Add the DataRaptor to the Integration Procedure.

- a. Return to the **Team Save Primary Contact Details** Integration Procedure.
- b. Select the **DRCreatPriContact** element.
- c. In the **DataRaptor Interface** dropdown, select your new DataRaptor (**teamCreatePriContact**).

Your Integration Procedure still needs two additional DataRaptors: one to Update the Primary Contact record with new / updated data and one to Change the Primary Contact from one Contact to another Contact.

Task 3: Build a DataRaptor Load to save updated primary contact data

1. Prepare the data input to use for creating the DataRaptor to update the Primary Contact.



To create the DataRaptor, you only need a portion of the JSON you pasted into the Preview: the particular node of the array.

- a. Open the **Team Save Primary Contact Details** Integration Procedure.
- b. Click **PREVIEW**.
- c. In **Input Parameters** on the left-hand side, change the **RadioPriContact** value to **Update** and update the email address in the **BlkUpdatePriContact** node.
- d. Click **Execute**.
- e. Expand the **Debug Log** and locate **DRUpdatePriContactDebug** from the dropdown.
- f. Select and copy the data input (including the brackets) and save it to re-use below.

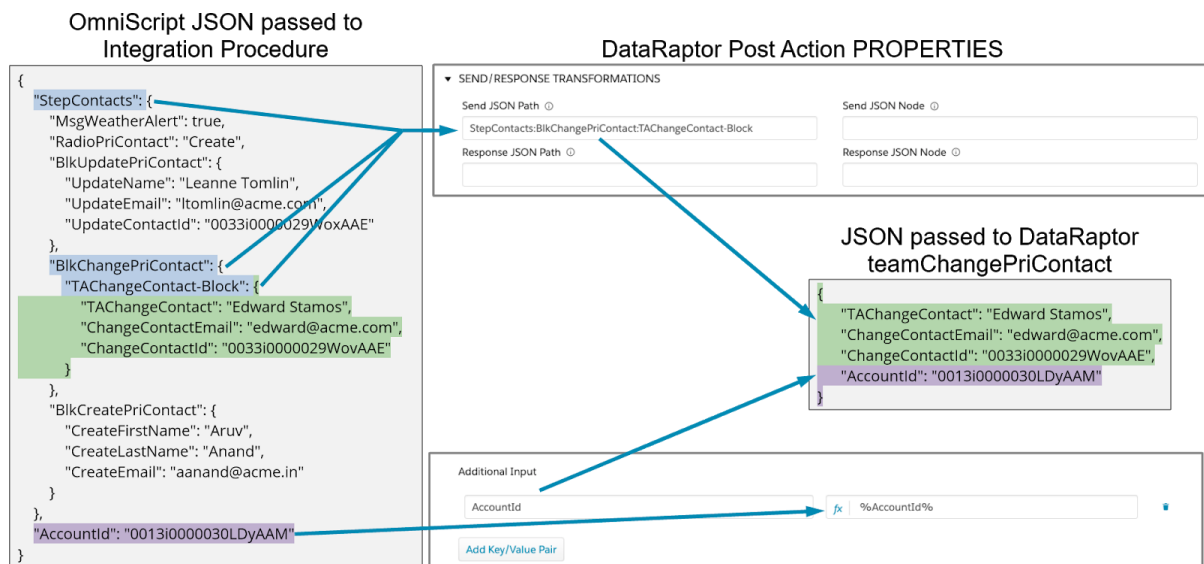
```
    "Input": {  
      "UpdateContactId":  
        "0034W00002Dz7rNQAR",  
      "UpdateName": "Leanne Tomlin",  
      "UpdateEmail":  
        "ltomlin2@acme.com"  
    }  
  }  
}
```

You are now ready to create the DataRaptor.

2. Build the new DRUpdatePriContact DataRaptor.
 - a. Open the **teamStarterUpdatePriContact** DataRaptor.
 - b. Click **Clone** and name the clone `teamUpdatePriContact` before saving.
 - c. The Contact is already mapped.
 - d. Click the **FIELDS** sub-tab.
 - e. Paste the JSON from the Integration Procedure into the **Input JSON** field.
 - f. Edit the **UpdateContactId** mapping to select **Upsert Key** and **Is Required For Upsert**.
3. Preview and test the new DataRaptor.
 - a. Click **PREVIEW**.
 - b. Paste the JSON from the Integration Procedure into the **Input** text box.
 - c. Edit some of the data and then, click **Execute**. There is one record in the **Objects Created** text box: a Contact.
 - d. Click the Contact link and confirm the email information you edited has been updated.

Task 4: Build a DataRaptor Load to save changed primary contact data

1. Prepare the data input to use for creating the DataRaptor to update the Primary Contact.



To create this DataRaptor, you only need a portion of the JSON you pasted into the Preview: the particular node of the array and the AccountId.

- a. Confirm you are viewing the **Team Save Primary Contact Details** Integration Procedure and click **PREVIEW**.
- b. In **Input Parameters** on the left-hand side, change the **RadioPriContact** value to **Change**.
- c. Click **Execute**.
- d. Expand the **Debug log** and locate **DRChangePriContactDebug** from the dropdown.
- e. Select and copy the data input (including the brackets) and save it to re-use below.

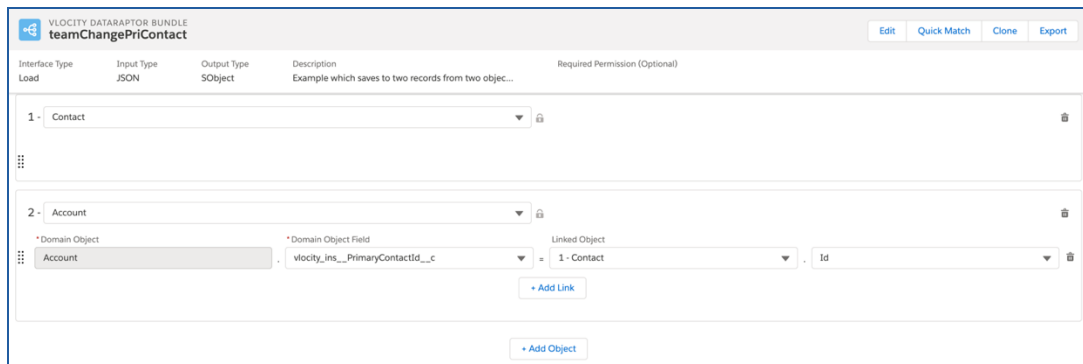
```

DRChangePriContactDebug
{
  "Status": true,
  "Type": "DataRaptor Post Action",
  "ElapsedTime": 7,
  "ElapsedTimeCPU": 5,
  "Input": {
    "AccountId": "00145000003MwEpQAK",
    "ChangeContactEmail": "edward@acme.com",
    "ChangeContactPhone": "(212) 169-1475",
    "ChangeContactId": "00345000002RkIgQAK",
    "ChangeContactNameKey": "Edward Stamos",
    "TAChangeContact": "Edward Stamos"
  }
}

```

You are now ready to create the DataRaptor.

2. Build the new DRChangePriContact DataRaptor.
 - a. Open the **teamStarterChangePriContact** DataRaptor.
 - b. Click **Clone** and name the clone **teamChangePriContact** before saving.
 - c. The Contact is already mapped, as is the Account, but you need to link the Account to the newly selected Contact.
 - d. Under **2-Account**, click **+ Add Link**.
 - e. In the **Domain Object Field**, select **vlocity_ins__PrimaryContactId__c**.
 - f. In the two **Linked Object** dropdowns, select **1-Contact** and **Id**



- g. Click the **FIELDS** sub-tab.
- h. Paste the JSON from the Integration Procedure into the **Input JSON** field.
- i. Select the **Contact** sub-tab.
- j. Edit the mapping for **ChangeContactId** as follows:

Property	Value
Upsert Key	✓
Is Required For Upsert	✓

- k. Select the **Account** sub-tab.
- l. Edit the mapping for **AccountId** as follows:

Property	Value
Upsert Key	✓
Is Required For Upsert	✓

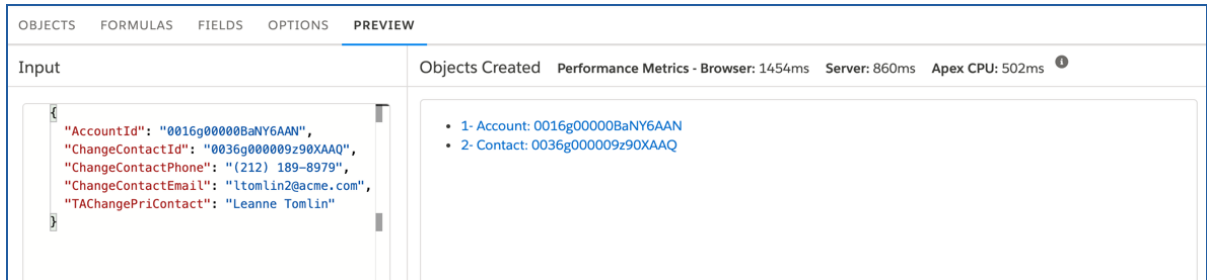
- m. Click the + to create a new mapping as follows:

INPUT JSON PATH	DOMAIN OBJECT FIELD	Note
ChangeContactId	vlocity_ins__PrimaryContactId__c	ChangeContactId is being mapped on both the Contact and Account sub-tabs.

OBJECTS	FORMULAS	FIELDS	OPTIONS	PREVIEW
1 - Contact	2 - Account	All		
INPUT JSON PATH 🔍		DOMAIN OBJECT FIELD 🔍		
ChangeContactEmail		1 - Contact.Email		✎ 🗑
ChangeContactId		1 - Contact.Id	⬆	✎ 🗑
AccountId		2 - Account.Id	⬆	✎ 🗑
ChangeContactId		2 - Account.vlocity_ins__PrimaryContactId__c		✎ 🗑
		2 - Account.vlocity_ins__PrimaryContactId__c	🔗	✎ 🗑

3. Preview and test the DataRaptor.
- a. Click **PREVIEW**.

- b. Paste the JSON from the Integration Procedure into the **Input** text box.
- c. Click **Execute**.
- d. There are two records in the **Objects Created** text box: an Account and a Contact.



The screenshot displays the OmniStudio interface with the **PREVIEW** tab selected. The **Input** section on the left contains a JSON payload:

```
"AccountId": "0016g00000BaNY6AAN",  
"ChangeContactId": "0036g000009z90XAAQ",  
"ChangeContactPhone": "(212) 189-8979",  
"ChangeContactEmail": "ltomlin2@acme.com",  
"TChangePriContact": "Leanne Tomlin"
```

The **Objects Created** section on the right shows the results of the execution:

- 1- Account: 0016g00000BaNY6AAN
- 2- Contact: 0036g000009z90XAAQ

Performance metrics are also visible: Browser: 1454ms, Server: 860ms, Apex CPU: 502ms.

- e. Click **Account** and confirm on the Acme details tab that the contact from the Input is now the **Primary Contact**, then close the browser tab.