

# 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

- none

## Tasks

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

## Time

- 30 mins

## Task 1: Prepare JSON with Conditional Inputs

Before you build an Integration Procedure with conditional DataRaptor Loads that can write any of these three changes back into your org, you need to gather the JSON you need to populate the different conditional inputs.

1. Populate the OmniScript JSON with the Update Contact inputs and save it for later.
  - a. Open the **App Launcher**, and select the **OmniStudio** app.
  - b. Click the dropdown arrow to open the menu and select **OmniScripts**.
  - c. Locate and access the **Sample Update Account Primary Contact** OmniScript.
  - d. In the Header, click **Preview**.
  - e. Paste the **Acme RecordId** into the **Context ID** field and refresh the preview.



### NOTE:

If you don't already have it, locate Acme's Record Id as follows:

1. Select **Accounts** from the Object dropdown.
2. If needed, switch the view to **All Accounts**.
3. Click **Acme** to view Acme's detail page.
4. Copy the RecordId from the URL (Account Ids always begin with 001, Contact Ids with a 003) and paste it somewhere to use it again.
5. Return to the DataRaptor and paste the RecordId into the **Value** field.

- f. Click **Action Debugger > Clear Logs**.
- g. Select the **Change primary contact** radio button in the preview screen.
- h. Click **Next**. This triggers the Integration Procedure that saves the data.
- i. In the **Action Debugger**, expand **IPSavePriContactDetails > Request Data**.
- j. Select and copy all the data in the text box and paste it in a file for use later – label it `Update JSON`. (You can also click the clipboard icon to copy all of the data.)



2. Populate the OmniScript JSON with the Change Contact inputs and copy it to save for later.
  - a. In the Action Debugger window, click **Clear Logs**.
  - b. Click **Refresh**.
  - c. Select the **Change primary contact** radio button in the preview screen.
  - d. Enter e in the name field and select **Edward Stamos**.
  - e. Click **Next**.

- f. Expand **IPSavePriContactDetails > Request Data**.
- g. Select and copy all the data in the text box and paste it in a file for use later – label it **Change JSON**.



```
{
  "StepContacts": {
    "BlkUpdatePriContact": null,
    "BlkChangePriContact": {
      "TChangeContact-Block": {
        "TChangeContact": "Edward Stamos",
        "ChangeContactNameKey": "Edward
Stamos",
        "ChangeContactLastName": "Stamos",
        "ChangeContactFirstName": "Edward",
        "ChangeContactId":
"0035f00000DUcqUAAI",
        "ChangeContactEmail":
"edward2@acme.com"
      }
    },
    "BlkCreatePriContact": null
  },
  "RadioPriContact": "Change",
  "AccountId": "0015f00000GRcuJAAD"
}
```

- 3. Populate the OmniScript JSON with the Create Contact inputs and copy it to save for later.
  - a. In the Action Debugger window, click **Clear Logs**.
  - b. Click **Refresh**.
  - c. Select the **Add a new primary contact** radio button and fill in the fields as follows:

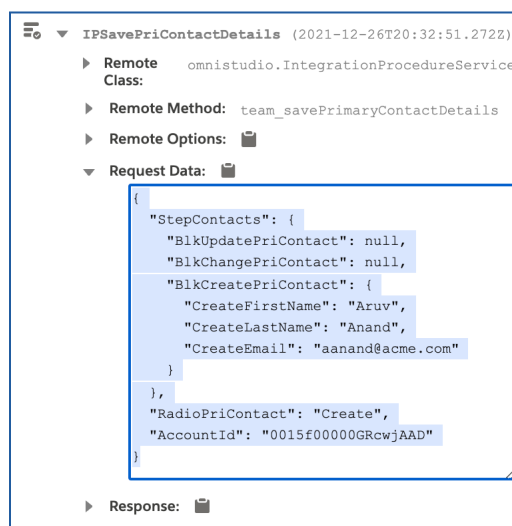
Property	Value
New Contact First Name	Aruv

---

New Contact Last Name	Anand
New Contact Email	aanand@acme.in

---

- d. Click **Next**.
- e. Expand **IPSavePriContactDetails > Request Data**.
- f. Select and copy all the data in the text box and paste it in a file for use later – label it `Create JSON`.



```
{
  "StepContacts": {
    "BlkUpdatePriContact": null,
    "BlkChangePriContact": null,
    "BlkCreatePriContact": {
      "CreateFirstName": "Aruv",
      "CreateLastName": "Anand",
      "CreateEmail": "aanand@acme.com"
    }
  },
  "RadioPriContact": "Create",
  "AccountId": "0015f00000GRwjAAD"
}
```

You now have JSON populated for all three of the conditional inputs.

## Task 2: 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**.

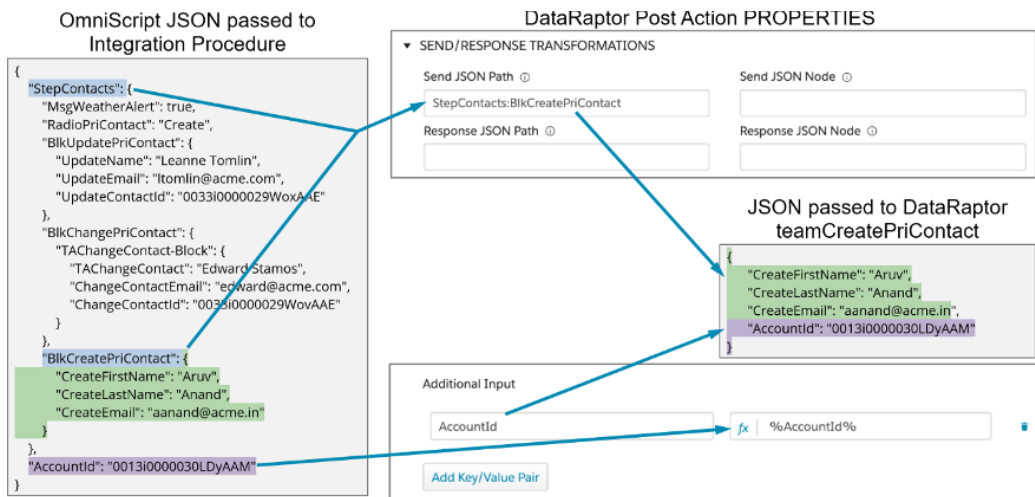
2. Confirm the conditions for when each of the three DataRaptors will execute.

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**, confirm the following:

Element	Field	Value
DRUpdatePriContact	Execution Conditional Formula	RadioPriContact=="Update"
DRChangePriContact	Execution Conditional Formula	RadioPriContact=="Change"
DRCreatePriContact	Execution Conditional Formula	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: the particular node of the array and the AccountId.

- Confirm you are still on the **DRCreatPriContact** element.
- Under **Additional Input**, click **Add Key/Value Pair**.
- In **Additional Input**, enter `Accountid` and in the **fx** field, enter `%Accountid%`.



**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 > Edit as JSON**.
- In the **Input Parameters** text box, if there is any data, click **Clear Data**, then paste the JSON you copied in Task 1 Step 3 above (the **Create** branch).

```
Input Parameters Edit as Params  
  
{  
  "StepContacts": {  
    "BlkUpdatePriContact": null,  
    "BlkChangePriContact": null,  
    "BlkCreatePriContact": {  
      "CreateFirstName":  
"Aruv",  
      "CreateLastName":  
"Anand",  
      "CreateEmail":  
"aanand@acme.com"  
    }  
  },  
  "RadioPriContact": "Create",  
  "AccountId": "0015f00000GRcwjAAD"  
}
```

- 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.

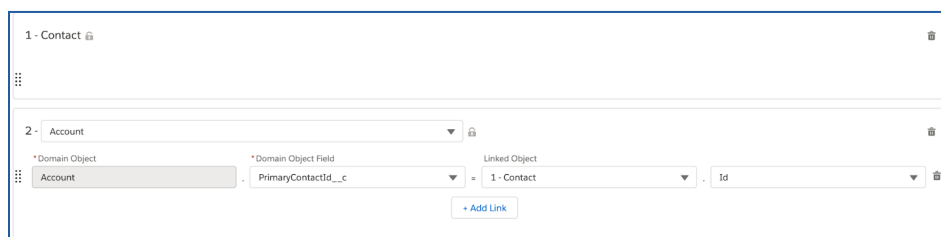
```
DRCREATEPRICONTACTDEBUG  
  
{  
  "Status": true,  
  "Type": "DataRaptor Post Action",  
  "ElapsedTime": 6,  
  "ElapsedTimeCPU": 4,  
  "Input": {  
    "AccountId": "0016g000003d0fbAAE",  
    "CreateEmail": "aanand@acme.in",  
    "CreateLastName": "anand",  
    "CreateFirstName": "aruv" 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

- i. You are now ready to build the Create DataRaptor.



## Task 3: 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 **PrimaryContactId\_\_c**.
  - f. In the two **Linked Object** dropdowns, select **1-Contact** and **Id**.



### 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.

You want to create a new Contact record and update Account information at the same time.



**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 and mark them as an **Upsert Key** (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.

- i. Confirm you are on the **Contact** sub-tab of the **FIELDS** sub-tab.
- j. Edit the **CreateLastName** mapping to select **Is Required For Upsert** and **Upsert Key**.
- k. Edit the **CreateFirstName** mapping to select **Upsert Key**.
- l. Click the **Account** sub-tab and click the **+** to add a new mapping.
- m. Fill in the fields as follows:

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

Because you need to update the existing Account record, adding an upsert key is required. DataRaptors automatically select AccountId as upsert keys. No need to mark them as such. Another example of the power of DataRaptors!

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.PrimaryContactId__c		🔗 ✎ 🗑

## 2. Preview and test the DataRaptor.

- Click **PREVIEW**.
- Below the **Input** text box, click **Clear Data** to pull the JSON data you pasted into the **Input JSON** field on the FIELDS sub-tab. This ensures the data you use to test matches the data you used to create the mappings.
- Click **Execute**.
- There will be 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"><li>1 - Account: 0013i000006rhJdAAI</li><li>2 - Contact: 0033i000003dWR8AAM</li></ul>		

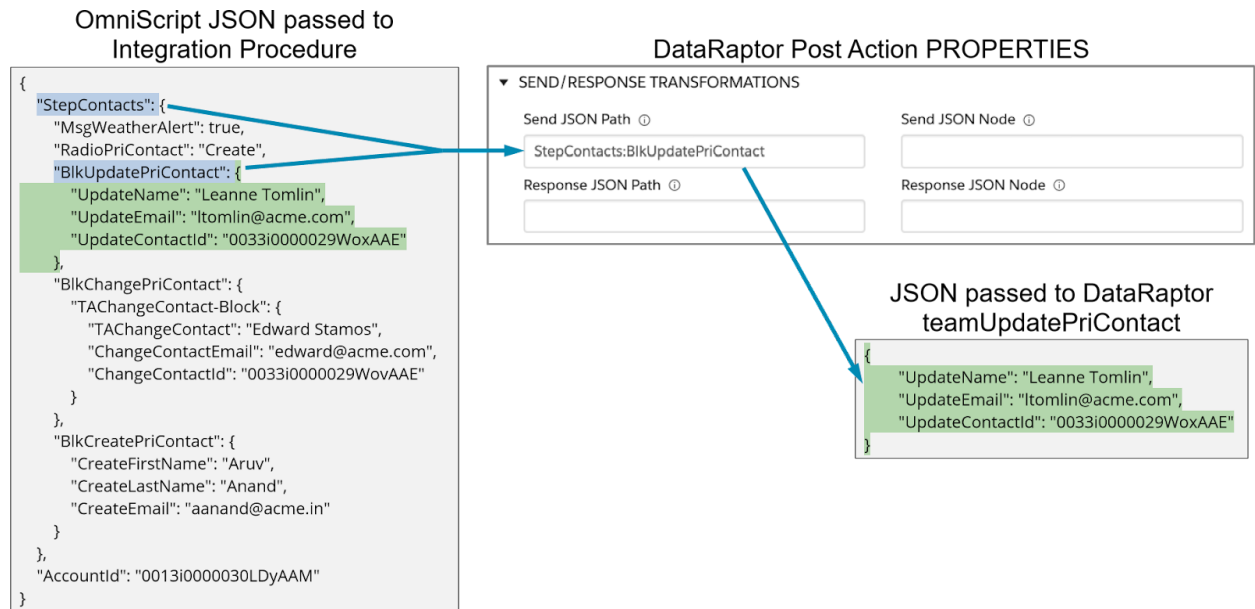
- Click the **Contact** link and confirm that **Aruv Anand** has been created and linked to **Acme** correctly.
- Click **Account** and confirm in the Header that **Aruv Anand** is now listed as the **Primary Contact**, then close the browser tab.

3. Add the DataRaptor to the Integration Procedure.
  - a. Return to the **Team Save Primary Contact Details** Integration Procedure.
  - b. Click **PROPERTIES** (if needed) and select the **DRCREATEPriContact** 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 4: 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 need to update the JSON used in Preview with that of the Update node.

- a. Open the **Team Save Primary Contact Details** Integration Procedure.
- b. Click **PREVIEW**.
- c. In **Input Parameters** on the left-hand side, paste the JSON for the Update branch, replacing the JSON that is already there.
- d. Update the email address.
- e. Click **Execute**.
- f. Expand the **Debug Log** and locate **DRUpdatePriContactDebug** from the dropdown.
- g. Select and copy the data input (including the brackets) and save it to re-use below.

```
"ElapsedTimeCPU": 127,  
"Input": {  
  "UpdateFirstName": "Aruv",  
  "UpdateLastName": "Anand",  
  "UpdateContactId":  
"0035f0000DMWgcAAH",  
  "UpdateEmail": "aanand@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 **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. 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 5: 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.

OmniScript JSON passed to Integration Procedure

```

{
  "StepContacts": {
    "MsgWeatherAlert": true,
    "RadioPriContact": "Create",
    "BlkUpdatePriContact": {
      "UpdateName": "Leanne Tomlin",
      "UpdateEmail": "ltomlin@acme.com",
      "UpdateContactId": "0033i0000029WovAAE"
    },
    "BlkChangePriContact": {
      "TChangeContact-Block": {
        "TChangeContact": "Edward Stamos",
        "ChangeContactEmail": "edward@acme.com",
        "ChangeContactId": "0033i0000029WovAAE"
      }
    },
    "BlkCreatePriContact": {
      "CreateFirstName": "Aruv",
      "CreateLastName": "Anand",
      "CreateEmail": "aanand@acme.in"
    }
  },
  "AccountId": "0013i0000030LDyAAM"
}
                
```

DataRaptor Post Action PROPERTIES

SEND/RESPONSE TRANSFORMATIONS

Send JSON Path

Response JSON Path

Send JSON Node

Response JSON Node

JSON passed to DataRaptor teamChangePriContact

```

{
  "TChangeContact": "Edward Stamos",
  "ChangeContactEmail": "edward@acme.com",
  "ChangeContactId": "0033i0000029WovAAE",
  "AccountId": "0013i0000030LDyAAM"
}
                
```

Additional Input

AccountId

[Add Key/Value Pair](#)

To create the DataRaptor, you need to update the JSON used in Preview with that of the Update node and confirm the appropriate information is being sent to the DataRaptor.

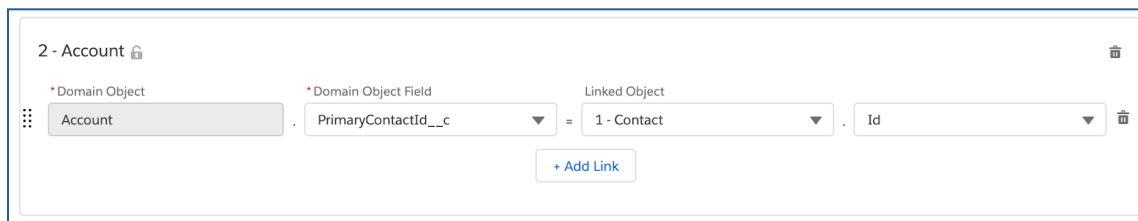
- a. Confirm you are viewing the **Team Save Primary Contact Details** Integration Procedure.
- b. Select the **DRChangePriContact** element.
- c. Under **Additional Input**, click **Add Key/Value Pair**.
- d. In **Additional Input**, enter `AccountId` and in the `fx` field, enter `%AccountId%`.
- e. Click **PREVIEW**.
- f. In **Input Parameters** on the left-hand side, replace the JSON with the JSON from the Change branch.
- g. Click **Execute**.

- h. Expand the **Debug log** and locate **DRChangePriContactDebug** from the dropdown.
- i. Select and copy the data input (including the brackets) and save it to re-use below.

```
DRChangePriContactDebug
{
  "Status": true,
  "Type": "DataRaptor Post Action",
  "ElapsedTime": 283,
  "ElapsedTimeCPU": 132,
  "Input": {
    "AccountId": "0015f00000GRcwjAAD",
    "ChangeContactEmail":
"lTomlin1@acme.com",
    "ChangeContactId":
"0035f00000DucqVAAT",
    "ChangeContactFirstName": "Leanne",
    "ChangeContactLastName": "Tomlin",
    "ChangeContactNameKey": "Leanne
Tomlin",
    "TChangeContact": "Leanne Tomlin"
  }
}
```

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 **PrimaryContactId\_\_c**.
  - f. In the two **Linked Object** dropdowns, select **1-Contact** and **Id**

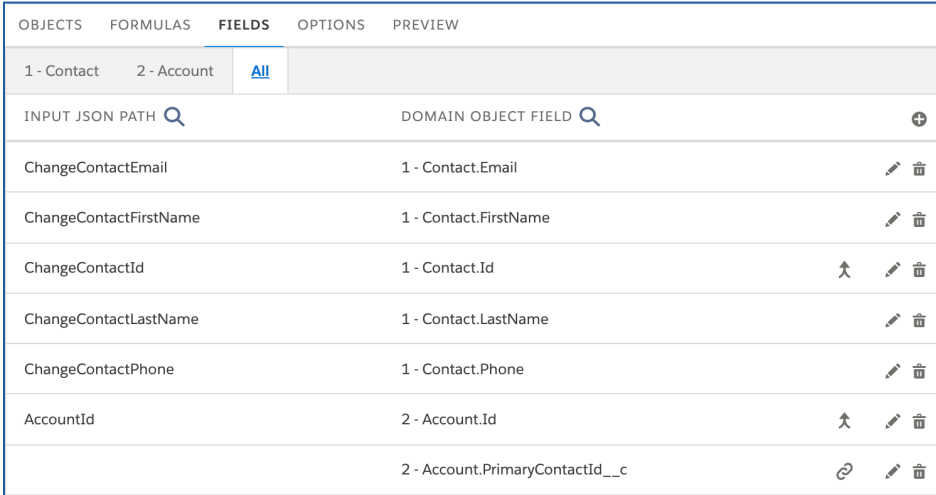


- g. Click the **FIELDS** sub-tab.



- 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** to select **Is Required For Upsert**.
- k. Select the **Account** sub-tab.
- l. Edit the mapping for **AccountId** to select **Is Required For Upsert**.
- m. Click the **+** to create a new mapping as follows:

INPUT JSON PATH	DOMAIN OBJECT FIELD	Note
ChangeContactId	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		
ChangeContactFirstName		1 - Contact.FirstName		
ChangeContactId		1 - Contact.Id	↑	
ChangeContactLastName		1 - Contact.LastName		
ChangeContactPhone		1 - Contact.Phone		
AccountId		2 - Account.Id	↑	
		2 - Account.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.

OBJECTS FORMULAS FIELDS OPTIONS PREVIEW

Input

```
"AccountId": "0015f00000GRcwjAAD",  
"ChangeContactEmail":  
"ltomlin1@acme.com",  
"ChangeContactId":  
"0035f00000DUcqVAAT",  
"ChangeContactFirstName": "Leanne",  
"ChangeContactLastName": "Tomlin",  
"ChangeContactNameKey": "Leanne  
Tomlin",  
"TAChangeContact": "Leanne Tomlin"
```

Objects Created Performance Metrics - Browser: 808ms Server: 432ms Apex CPU: 224ms

- 1- Contact: 0035f00000DUcqVAAT
- 2- Account: 0015f00000GRcwjAAD

- 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.