Before You Begin

Did you sign up for a special OmniStudio Developer Edition org already? You'll need one to do the steps in this guide. Here's how to request one if this is your first time completing an OmniStudio module:

1. Sign up for a [free Developer Edition org with OmniStudio](#).
2. Fill out the form.
   a. For Email, enter an active email address.
   b. For Username, enter a username that looks like an email address and is unique, but it doesn't need to be a valid email account (for example, [yourname@omnistudiotrails.com](#)).
   c. After you fill out the form, click **Sign me up**. A confirmation message appears.
3. When you receive the activation email (this might take about 10 minutes), open it and click **Verify Account**.
4. Complete your registration by setting your password and security challenge question. **Tip:** Write down your username, password, and login URL for easy access later.

You are logged in to your Developer Edition and you can begin practicing.

Extract Data from Multiple Salesforce Objects

Requirements

"I'd like the Update Account Primary Contact OmniScript to extract data from Salesforce."

Create a DataRaptor Extract that pulls data from two sObjects, then link this DataRaptor to an Integration Procedure that you use to update the Update Account Primary Contact OmniScript.

The DataRaptor Extract must retrieve Primary Contact information from both an Account and Contact. (Primary Contact is an OmniStudio custom field in the Account object).

Prerequisites

- None

Tasks

1. Build a DataRaptor Extract that Gets Data from a Parent sObject
2. Get Data from a Parent sObject with an Integration Procedure

Time

- 20 mins

Task 1: Build a DataRaptor to Get Data from Multiple sObjects

1. Create a DataRaptor Extract.

   a. Open the **App Launcher**, and select the **OmniStudio** app.

   b. Click the dropdown arrow to open the menu and select **OmniStudio DataRaptors**.

   c. Click **New.**

   d. Enter the following information to create the DataRaptor:
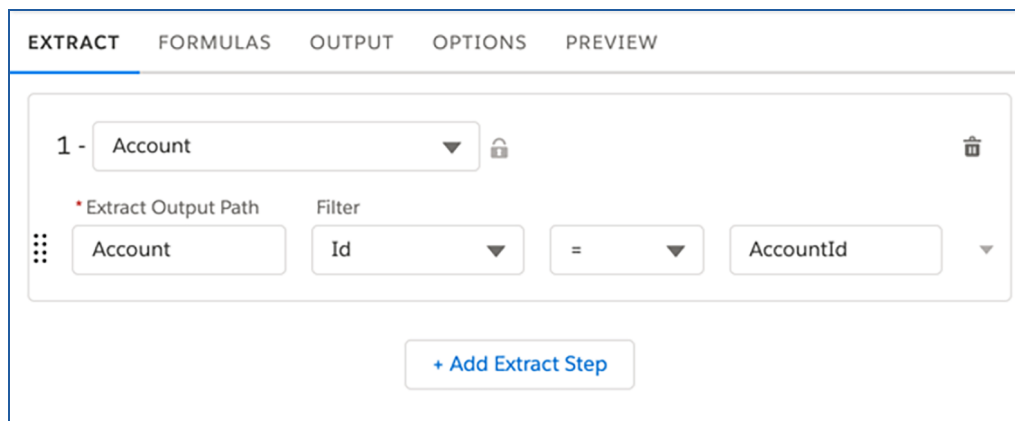
Property	Value	Notes
DataRaptor Interface Name	`teamGetPriContactDetails`	Best practice naming conventions for DataRaptors follow **prefixVerbObjectDetail**, use camelCase to facilitate reading, and begin with a lowercase letter for Lightning Web Component compatibility.
Interface Type	Extract	There are four types of DataRaptors: **Extract** (to get data from Salesforce), **Turbo Extract** (to retrieve data from a single Salesforce object type), **Load** (to write data back into the database) and **Transform** (transform or convert data from one structure to another).

Input Type	JSON	Click to review the different output types but leave as JSON.
Output Type	JSON	Click to review the different output types but leave as JSON.

    d. Click **Save**.

2. Define the objects you are extracting data from.

    a. Confirm you are on the **EXTRACT** sub-tab.

    b. Click **+ Add Extract Step**.

    c. In the top field, select **Account**.

    d. In the **Extract Output Path**, enter `Account`. This is the name of the incoming JSON node the DataRaptor needs to find.

    e. In the **Filter** field, select **Id**. This is the field used to filter the results, which in this case is an AccountId.

    f. In the final field, enter `AccountId`. This is the name of the key in the incoming JSON.

3. Define the Output path for your DataRaptor.

   a. Select the **OUTPUT** sub-tab.

   b. In the upper right-hand corner, click the ⬚ (grey +) to add an extract path and continue as needed to create the following:

EXTRACT JSON PATH	OUTPUT JSON PATH	Notes
Account:BillingPostalCode	AccountPostalCode	
Account:Id	AccountId	
Account:vlocity_ins__Prim aryContactId__c	UpdateContactId	

> **NOTE:**
> The EXTRACT JSON PATH dropdown displays every field available on the Account record (the object we are extracting data from). Start typing to filter the list. For example, enter `Billing` to quickly locate **Account:BillingPostalCode.**
>
> The OUTPUT JSON PATH defines the way the data displays in the output JSON node.

   c. Review the **Current JSON Output** and confirm the node has three fields.

4. Edit the DataRaptor Output to add relationship queries.
   a. From the Primary Contact mapping, copy the value of the **EXTRACT JSON PATH** (**Account:vlocity_ins__PrimaryContactId__c**).

   b. Click the grey **+** to create a new mapping.

   c. Paste the value you copied in the **EXTRACT JSON PATH** field.

   d. Replace the **c** at the end of the field name with `r.Name`.

      e. For the **OUTPUT JSON PATH**, select **UpdateName**.

> **TIPS:**
> The r. allows you to create a relationship query within the SOQL query of the DataRaptor to pull in any field of the related Contact sObject, in this case the contact name.

      f. Add a new mapping and paste the value you copied into the new **EXTRACT JSON PATH** field.

      g. Edit the name to add `r.Email`.

      h. For the **OUTPUT JSON PATH**, select **UpdateEmail**.

      i. Review the updated mappings that now include your two relationship queries.

EXTRACT JSON PATH 🔍	OUTPUT JSON PATH 🔍	⊕
Account:BillingPostalCode	AccountPostalCode	✏ 🗑
Account:Id	AccountId	✏ 🗑
Account:vlocity_ins__PrimaryContactId__c	UpdateContactId	✏ 🗑
Account:vlocity_ins__PrimaryContactId__r.Email	UpdateEmail	✏ 🗑
Account:vlocity_ins__PrimaryContactId__r.Name	UpdateName	✏ 🗑

5. Preview the data.

      a. Click **PREVIEW > Edit as Params > + Add New Key/Value Pair**.

      b. Enter the values as follows:

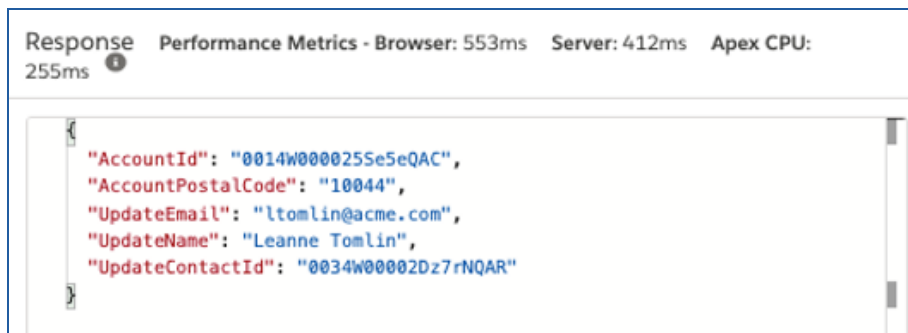| Property	Value
Key	`AccountId`
Value	[Acme account's RecordId]

> **NOTE:**
> If you don't already have it, locate Acme's Record Id as follows:
> 1. Duplicate the browser tab, then use the App Launcher to open the **OmniStudio Console**.
> 2. Select **Accounts** from the Object dropdown.
> 3. If needed, switch the view to **All Accounts**.
> 4. Click **Acme** to view Acme's detail page.
> 5. 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.
> 6. Return to the DataRaptor and paste the RecordId into the **Value** field.

    c. Click **Execute** and confirm the **Postal Code** and **AccountId** for Acme and **Contact** information for Leanne Tomlin displays.

Response   Performance Metrics - Browser: 553ms   Server: 412ms   Apex CPU: 255ms ⓘ

```
{
    "AccountId": "0014W000025e5eQAC",
    "AccountPostalCode": "10044",
    "UpdateEmail": "ltomlin@acme.com",
    "UpdateName": "Leanne Tomlin",
    "UpdateContactId": "0034W00002Dz7rNQAR"
}
```

Task 2: Get Data from a Parent sObject with an Integration Procedure

1. Add the DataRaptor to the Integration Procedure.

   a. From the tools dropdown list, select **OmniStudio Integration Procedures**.

   b. Open **team/getPrimaryContactDetails > Team Starter Get Primary Contact Details (Version 1)**.

   c. Click **Create Version**.

   d. Remove **Starter** from the name. The new name is **Team Get Primary Contact Details**. Because of the Starter and Stub versions, this is version 3.

   e. In the STRUCTURE panel, select the **DataRaptor Extract Action** element **DRGetPriContactDetails** (below Procedure Configuration).

   f. Click in the **DataRaptor Interface** field and select **teamGetPriContactDetails**.

      The input parameters and Response Action have already been set.

   g. Click **PREVIEW**.

   h. Paste the Acme RecordId into the value field.

   i. Click **Execute** and confirm the Primary Contact data displays.

j.   Click **PROPERTIES > Procedure Configuration > Activate Version**.

2. Confirm the active Integration Procedure is connected to your OmniScript.

a.   From the tools dropdown list, select **OmniScripts**.

b.   Locate and open the **Sample Update Account Primary Contact 4-9** OmniScript.

c.   In the Structure panel, click **IPSavePriContactDetails**.

d.   In the Properties panel, edit the Integration Procedure as **team_savePrimaryContactDetails.**

e.   Click **Preview**.

f.   Paste **Acme's RecordId** into the **Context ID** field.

g.   Click **Refresh**.

h.   Confirm the Primary Contact data is visible in the OmniScript.

Review

Confirm your understanding by answering these questions.

1. What two sObjects are used to pull data for the Update Primary Account Contact OmniScript?

2. What did adding r.FIELDNAME to the end of the field names in the EXTRACT JSON PATH do?

3. Why did you preview the DataRaptor and your updated OmniScript?