



Map It 360 Admin Guide

Version 1.0, Winter '19

Table of Contents

Introduction	2
Getting Started	2
Map It 360 Admin Setup	4
Inputs Setup	5
1: Default Setup	6
2: Eligible Standard sObjects	7
2.1 Usage	9
3: Custom or Ineligible Standard sObjects	10
3.1 Usage	11
Additional Features	12
1: Where Clause	13
2: Radius	15
Outputs Setup	17
Troubleshooting	18
Glossary	19
Acknowledgements	21

Introduction

View a collection of records that have an address field, in a map, which can be filtered, searched and selected using Stratus360's Map It 360 flow solution.

Available for any default, or custom sObject with address fields, Map It 360 is a versatile flow solution to visualize the locations of a specific set of locations. The set of records are dynamic, and specified by the admin. Map It 360 features a list of locations as well, which can be selected for further usage in subsequent flow screens.

The Map It 360 Admin Guide provides the setup process as well as assists in troubleshooting any problems the user might face.

Example Features:

- Customizing which sObject type gets shown
- Specifying a subset of a certain object to map
- Creating a custom object that is eligible to be used in Map It 360

For questions and support, contact us at:

info@stratus360.com
(866) 652-5818

Getting Started

Let's get familiar with the Map It 360 managed package.

The Map It 360 managed package includes:

- **Map It 360 Component**, which is the flow solution lightning component.
- **Location List Component**, which is the child component to the Map It 360 Component. It allows users to search through a list of locations that are marked on the map, and select a specific item.

Map It 360 Admin Setup

Map It 360 requires the administrator to complete one setup process before users can see Map It 360 in flows: specify the address fields.

Input Setup

- [Default Setup](#)
- [Eligible Standard sObject Setup](#)
- [Custom or Ineligible Standard sObjects](#)

Additional Features

- [Where Clause feature](#)
- [Radius feature](#)

Output Setup

- [Output Setup](#)

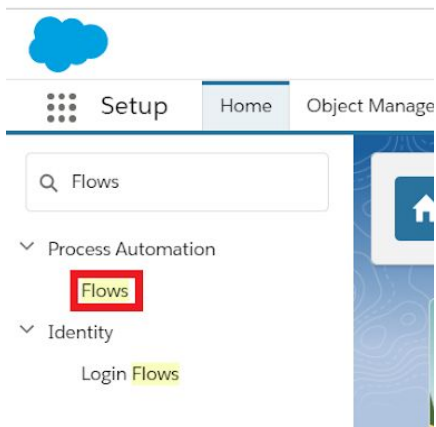
Inputs Setup

This setup is to be used when the sObject trying to be mapped are **Accounts**. The default address fields used to locate them on the map are:

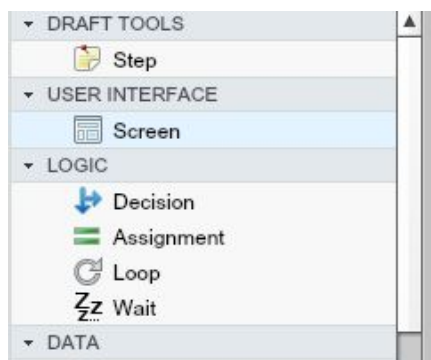
BillingStreet
BillingCity
BillingState
BillingCountry
BillingPostalCode
BillingLongitude
Billing Latitude

1: Default Setup

1. Search “**Flows**” in Setup and click on “**Flows**”.



2. Click “**New Flow**” in to open up the Flow Designer.
3. Under the **User Interface** tab on the left, drag a new **Screen** into the Flow Designer.

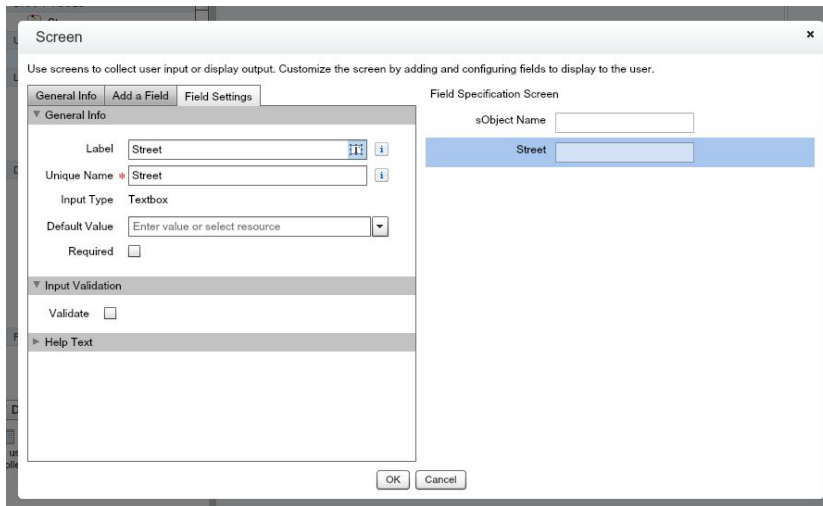


4. Click the  icon on the flow screen, to edit it.
5. Under the **Add a Field** tab, drag a **Lightning Component** to the Screen on the right. Click **[Lightning Component]**
6. In the **Field Settings** tab, give the field a name, and in the **Lightning Component** dropdown, select “**S360_AM:AddressMapper**”. Make sure that there are no inputs or outputs. This flow screen, for the purposes of this guide, is called the *Map It 360 Screen*.
7. Click  on the top right of the Save the flow, and click **Run**. By default, Map It 360 will show all **Account** records with a valid address in the **BillingAddress** field.

2: Eligible Standard sObjects

This setup is used for sObjects with Address-type fields by default. This does not include **Account** via **BillingAddress**, which can be mapped out using [1: Default Setup](#). Eligible standard sObjects are **Account** (using **ShippingAddress**, rather than **BillingAddress**), **Contact**, and **Lead**.

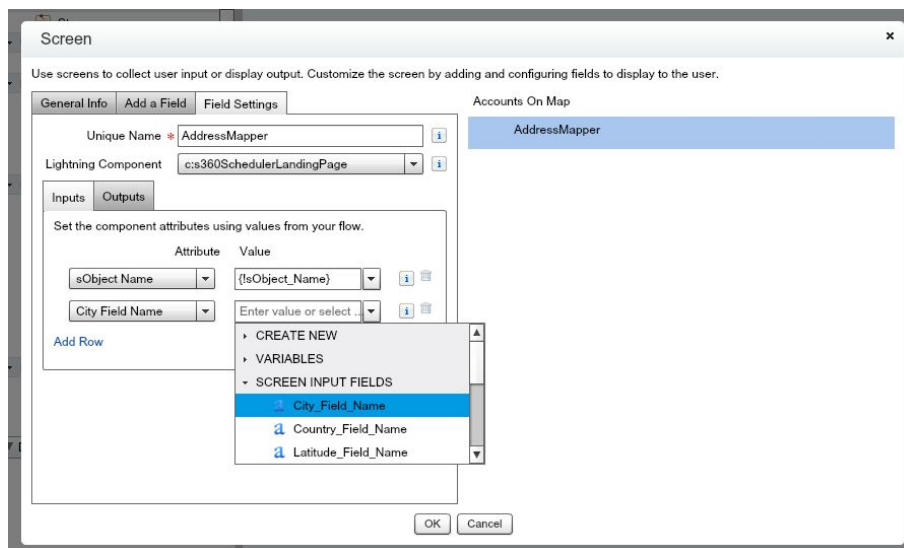
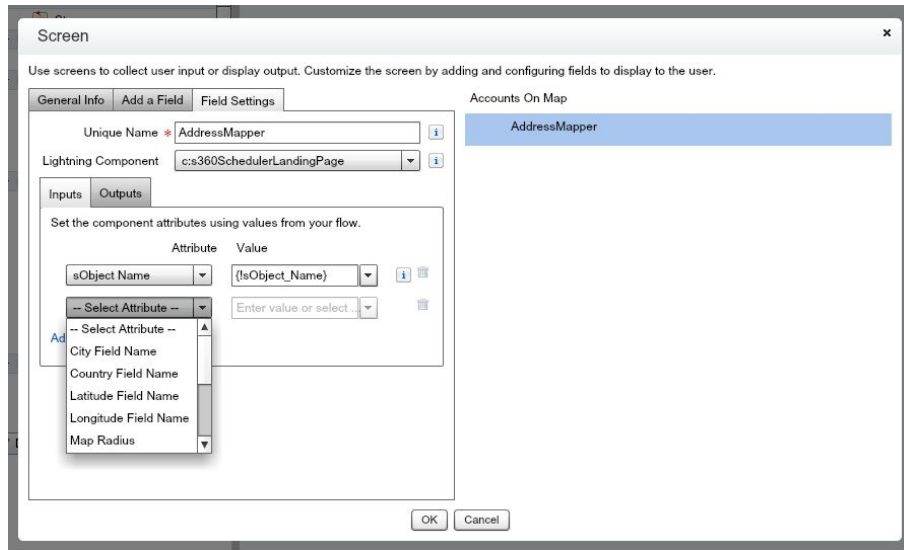
1. Do steps 1 - 7 from the [1: Default Setup](#) section above.
2. Create a new screen in the flow, by dragging the **Screen** item from the **User Interface** tab.
3. Under **Add a Field**, add 8 textbox fields by dragging and dropping them into the flow screen. Label each of the fields **sObject Name**, **Street**, **City**, **State**, **Country**, **Postal Code**, **Latitude**, and **Longitude**. Click **OK**. This screen is called the *Field Specification Screen*.



4. Click the diamond below the *Field Specification Screen* (blue), and drag the arrow to the *Map It 360 Screen* (black).



5. Open the *Map It 360 Screen* and select the Lightning Component in the right screen. Under **Inputs**, click **Add Row** and select the attribute for each of the fields **sObject Name**, **Street**, **City**, **State**, **Country**, **Postal Code**, **Latitude**, and **Longitude**, with the **Value** being the corresponding textbox input field's Unique Name.



6. Click the “**OK**” button.

7. Click the  on the *Field Specification Screen*, click **Save**, and then **Run**.

2.1 Usage

When running Case 2: Eligible Standard sObjects, on the Field Specification Screen, make sure the fields are filled in properly. The field API names are the values that should be typed into the input fields. Ensure that the API names are spelt correctly.

sObject Name	Contact
Street Field Name	MailingStreet
City Field Name	MailingCity
State Field Name	MailingState
Country Field Name	MailingCountry
Postal Code Field Name	MailingPostalCode
Latitude Field Name	MailingLatitude
Longitude Field Name	MailingLongitude

3: Custom or Ineligible Standard sObjects

For custom sObjects, or standard sObjects which do not have an address field by default, address fields need to be created.

1. Follow steps **1 - 6** from the [2: Eligible Standard sObjects](#) section.
2. In **Setup -> Object Manager**, select the sObject to be mapped out. If it does not exist yet, click **New -> sObject** and follow the procedure to create a new sObject.
3. In the **Fields and Relationships** tab on the left, select **New**.
 - Create a text field for **Street**, **City**, **State**, **Country** and **Postal Code**. These are the address fields.
 - Create a geolocation-type field for the **Coordinates**

Details	Fields & Relationships 10 Items, Sorted by Field Label			Quick Find
Fields & Relationships	FIELD LABEL	FIELD NAME	DATA TYPE	
Page Layouts	Street	Street__c	Text(255)	
Lightning Record Pages	State	State__c	Text(255)	
Buttons, Links, and Actions	Postal Code	Postal_Code__c	Text(255)	
Compact Layouts	Owner	OwnerId	Lookup(User,Group)	
Field Sets	Last Modified By	LastModifiedById	Lookup(User)	
Object Limits	Incidents Name	Name	Text(80)	
Record Types	Created By	CreatedById	Lookup(User)	
Related Lookup Filters	Country	Country__c	Text(255)	
Search Layouts	Coordinates	Coordinates__c	Geolocation	
Triggers	City	City__c	Text(255)	
Validation Rules				

4. For each record fill in each of the address fields, and manually enter the coordinates fields (there is no automatic geolocation on these objects, unless it is manually written as a data integration rule)

3.1 Usage

When running [Case 3: Custom or Ineligible Standard sObjects](#), on the Field Specification Screen, make sure the fields API names are filled in properly. The field API names are the values that should be typed into the input fields. Coordinate fields for these sObjects need to be manually entered, as opposed to [Case 2: Eligible Standard sObjects](#) where geolocation is automatic.

For the Latitude and Longitude fields, the field name will be:

Fieldname__Latitude__s

Fieldname__Longitude__s

In the case with the image below, the coordinates field API name is **Coordinates**, so the field names are **Coordinates__Latitude__s** and **Coordinates__Longitude__s**.

sObject Name	Incident__c
Street Field Name	Street__c
City Field Name	City__c
State Field Name	State__c
Country Field Name	Country__c
Postal Code Field Name	Postal_Code__c
Latitude Field Name	Coordinates__Latitude__s
Longitude Field Name	Coordinates__Longitude__s

Additional Features

For use cases when trying to map out object types that are not Account, Map It 360 needs to be properly configured, and Admins need to specify the input fields for the flow.

Where Clause

The where clause allows for preliminary filtering before a query is made. This means that any records filtered out here will not show up on the map. It can also be used to specify a certain subset of an object type. It is written in the format of a [SOQL Conditional Expression](#).

Radius

The Radius feature allows to center the map around the user's current location. If a number is entered, the map will center around the user's location and provide a viewport with the radius specified (in km).

In this section, we will go through:

- [Adding a text field for the Where Clause](#)
- [Using the Where Clause feature](#)
- [Adding a number field for Radius](#)
- [Using the Radius feature](#)

1: Where Clause

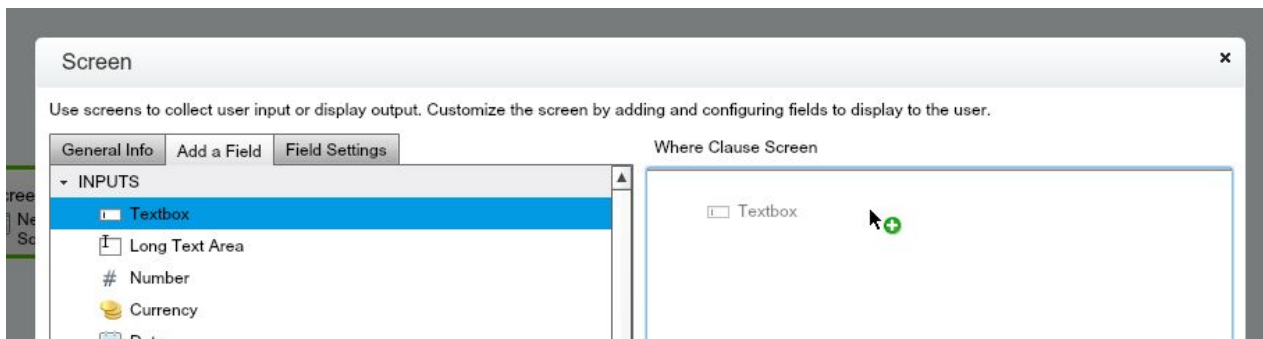
The where clause allows for preliminary filtering before a query is made. This means that any records filtered out here will not show up on the map. It can also be used to specify a certain subset of an object type. It is written in the format of a [SOQL Conditional Expression](#).

1. Follow steps 1-7 in [1: Default Setup](#).
2. In the flow, add a screen before the *Map It 360 Screen* created in Step (1). Give it a name.

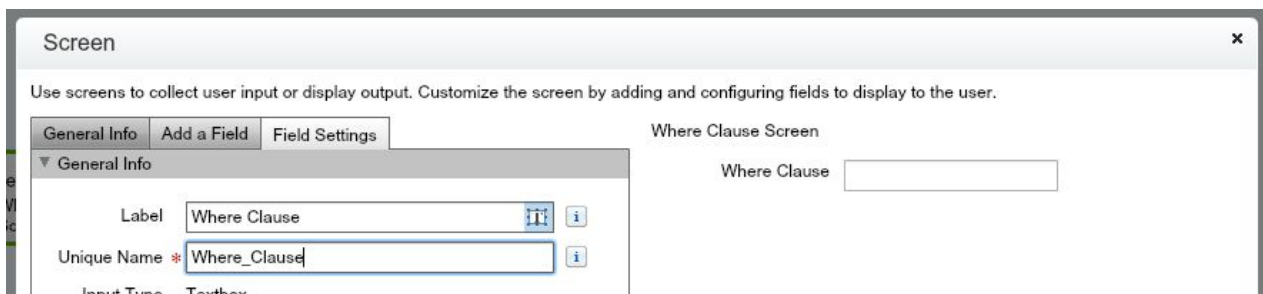


If a *Field Specification Screen* has already been created, you do not need to create a new flow screen. It can be used as the *Where Clause Screen*.

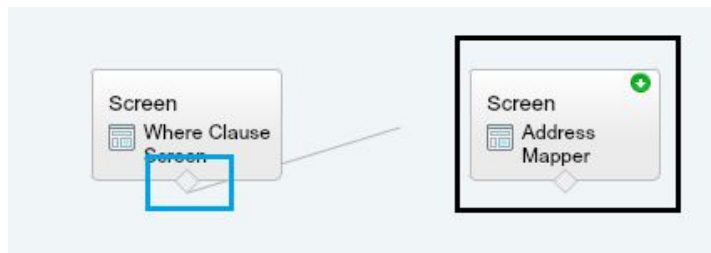
3. Under **Add a Field** tab, drag a **Textbox** to the screen editor on the right.



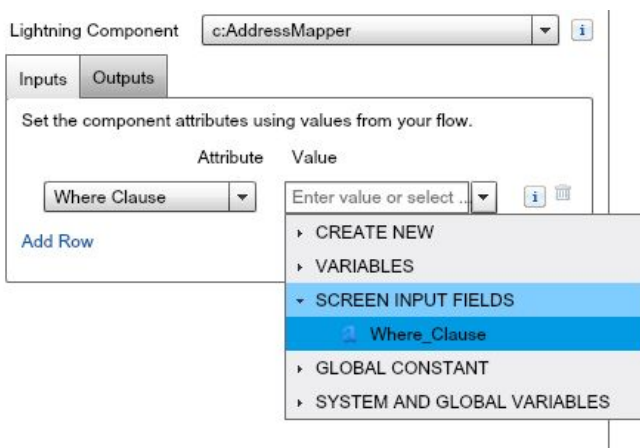
4. In **Field Settings** tab, give the textbox a name, and a unique name, and click **OK**.



5. Click the diamond below the Where Clause screen (Blue), and drag it to the Map It 360 Screen (Black).



6. In the Map It 360 screen editor, go to **Field Settings**, click **Add Row** under **SCREEN INPUT FIELDS**. Under Select Attribute, select **Where Clause**. In Value, select the unique Id of the textbox created in step 4. Click **OK**.



7. Click the  on the where clause screen (or the *Fields Specification Screen* if doing [Case 2](#) or [Case 3](#)) and **Save** and then **Run**.

USAGE:

When typing in a where clause, ensure the spelling is typed exactly how it would be in a **DML Statement**, not including the word 'WHERE'. Some examples are:

For all accounts with a billing address in Canada:
BillingCountry = 'Canada'

If using **Case 2**, with Lead, for any Lead where the State **IS NOT** New York (NY):
State != 'NY'

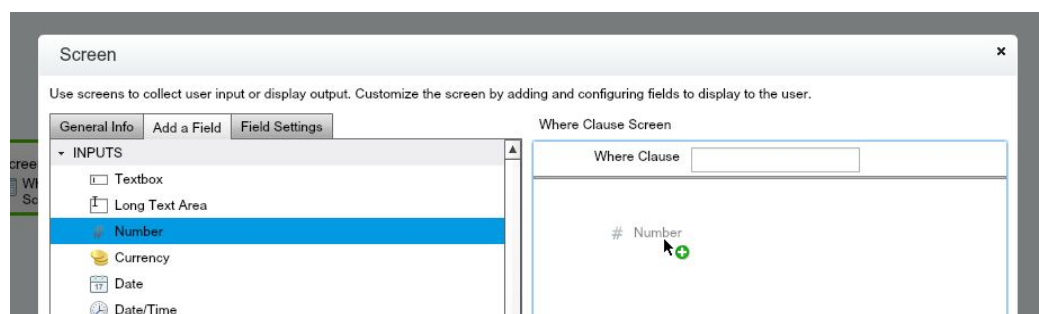


It is imperative to use regular single quotes (' ') and not curly single quotes (' ').

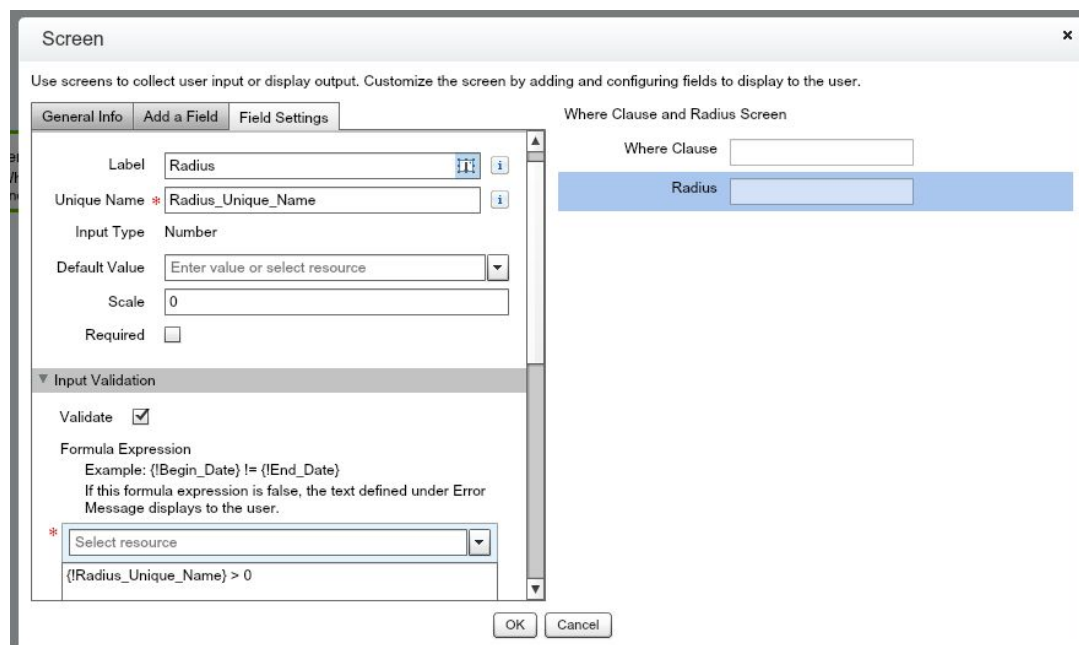
2: Radius

By default, the Map It 360 map will adapt the viewport to fit all markers inside the map. The Radius feature overrides this, and centers the map around the user's current location. If a number is entered, the map will center around the user's location and provide a viewport with the radius specified (in km). If left blank, it will return to the default setting of showing all markers.

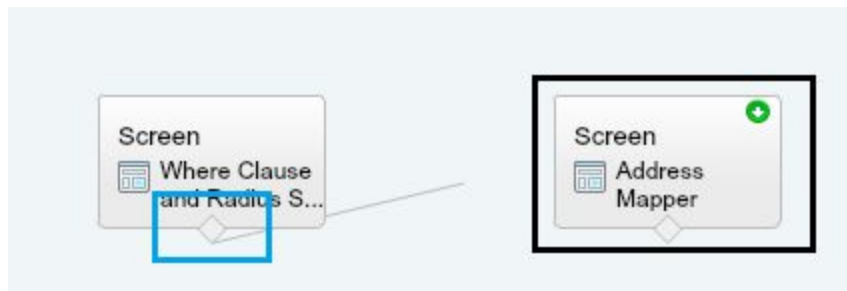
1. Follow Steps 1-2 of creating the [Where Clause](#) feature.
2. Under **Add a Field** tab, drag a **Number** field to the screen editor on the right.



3. In the **Field Settings** tab, give the Number field a label and a unique name. In the **Input Validation** section, check **Validate**. In the formula expression, type in "{!Unique_Name} > 0" where Unique_Name is replaced by the unique name given to the Number field.



4. Click the diamond below the Radius screen (Blue), and drag it to the Map It 360 Screen (Black).



5. In the Map It 360 screen editor, go to **Field Settings**, click **Add Row**, and under **Select Attribute**, select **Map Radius**. In value, under **SCREEN INPUT FIELDS**, select the unique Id of the Number field created in step 2. Click **OK**.

The screenshot shows the 'Screen' editor window with the 'Field Settings' tab selected. The 'Inputs' section is active, and the 'Add Row' button is clicked, opening a dropdown menu. The dropdown menu shows 'SCREEN INPUT FIELDS' and 'Radius_Unique_Name'.

Attribute	Value
Where Clause	{!Where_Clause}
Map Radius	Enter...

Add Row

- CREATE NEW
- SCREEN INPUT FIELDS
 - Radius_Unique_Name

6. Click the  on the radius screen (or the *Fields Specification Screen* if doing case 2 or 3) and **Save** and then **Run**.

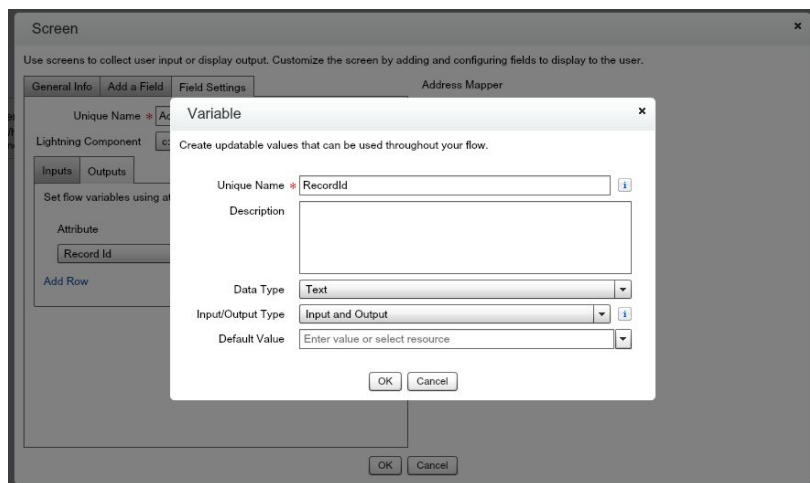


Enable location settings when using the Radius feature.

Outputs Setup

In the Map It 360 screen, there is a map, which shows all the locations of a specific object type, and a list of the locations. If an item in the list is selected, it can be outputted to the next flow screen for any manipulation. Other variables can also be outputted, but for the guide we will take a look at the id of the selected record.

1. Open the *Map It 360 Screen* editor, and, under **Field Settings**, go to **Outputs**.
2. Click **--Select Attribute--** and Select **Record Id**.
3. Click the **Value** dropdown, then click **CREATE NEW**, then **Variable**.
4. Give the variable a name, data type, and for **Input/Output Type** select **Input and Output**. Click **OK**, then click **OK**.



5. Now you have assigned a flow variable with the record Id of the selected record. If a user selects a record, and clicks the next button in the flow, the flow variable **RecordId**, will be populated with that Id.

Troubleshooting

The inputs need to be typed exactly how they are spelled in their API names, and the where clause needs to be spelled exactly how it would in a query. Addresses also need to be spelled exactly how they are in reality, to allow Salesforce's geocoding engine to properly find a coordinate pair.



Error 1: Error happened when processing action responses [Bounds are not valid. Callback failed: apex://AddressMapperController/ACTION\$getLocations] Failing descriptor: {markup://c:AddressMapper}

Reason: An input field has been spelled incorrectly, this can be in 2 places:

1. If a field name is incorrectly spelled
2. The where clause is not correctly structured

Solution: Make sure both the where clause (if used) or the field names are spelled exactly how they should be. Remember to not use the word 'WHERE' in the clause itself, and to use regular single quotation marks (' ') as opposed to curly ones (' ').

The image below would cause **Error 1**. The correct spelling of the field API name for a Lead's city is 'City', however it is spelled below as 'Citty'.

The image shows a Salesforce form with four input fields. The first field is labeled 'sObject Name' and contains 'Lead'. The second field is labeled 'Street Field Name' and contains 'Street'. The third field is labeled 'City Field Name' and contains 'Citty'; this field is highlighted with a red border. The fourth field is labeled 'State Field Name' and contains 'State'.



Error 2: No records are showing up, despite all spelling is correct.

Reason: Make sure the addresses are spelled correctly, and the coordinates (if using Case 3), are correctly formatted. If the addresses are misspelled, or incorrectly typed, Salesforce will fail to geocode them, and thus will fail to show up on Map It 360.

Solution: Ensure the addresses are correctly spelled in the record detail page

Glossary

A

Map It 360 - A flow solution that allows a user to place a set of records on a map, alongside a filterable list.

Acknowledgements

[Great Web Icons](#) created by Freepik.