Salesforce Workflow Connector

Download the PDF of this article.

In this Article

Related Articles

Overview

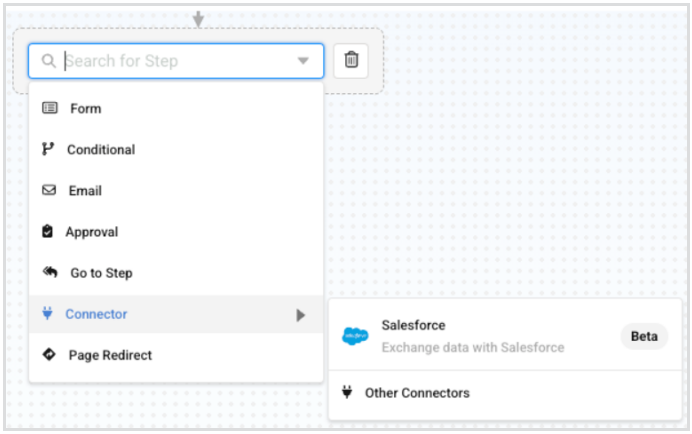
Send, receive, and prefill Salesforce data with the Salesforce Workflow Connector.

Requirements

To use the Salesforce Workflow Connector, a FormAssembly user must:

- be part of a Team plan or higher
- purchase the Salesforce Connector Add-on for their instance
- have Content Management and Salesforce user permissions enabled for their user account
- have a Salesforce account

Add the Salesforce Connector to your Workflow



- Select **Add Step** and choose **Connector**
- Select **Salesforce**

Note: The form-native version of the Salesforce connector can still be added to workflows by selecting **Other Connectors** in the **Add Step: Connector** menu. For more information on the previous Salesforce connector, begin with this document.

Note: If the user lacks the FormAssembly user permission to Salesforce but the Salesforce Workflow Connector step is configured in the workflow, the connector will be skipped.

Set up the Connector Properties

Refer to our Workflow-native Connectors article for general configuration information.

Authorization Tab

The Authorization dropdown lists all saved authorizations for forms and workflows owned by your user account. Select an existing authorization from the list or scroll to the bottom to create a New Authorization.

Create a New Authorization

- Select **New Authorization** from the Authorization dropdown list
- Enter a **Name** for the authorization
    - We advise using an easily recognizable name so that you'll know which Salesforce environment and login credentials the authorization belongs to
- Click **Authorize in Production** or **Authorize in Sandbox**
    - Select the appropriate authorization option based on the environment type you will use with the connector

Grant access to Salesforce

- Sign in with your Salesforce account credentials
    - For custom domains in Salesforce, select "**Use Custom Domain**" when logging in
- Accept the permissions requested by the Salesforce Workflow Connector for your Salesforce account
    - If you do not accept the permissions, you will be unable to use the connector
- Click **Save** to store and apply this authorization to the connector

Reauthorize Salesforce

To quickly reauthorize your existing Salesforce Connections from one place,

- Navigate to the External Authorizations page
    - Click **Manage** within the Authorization tab, or
    - Select the **Preferences Menu > External Authorizations**.
- Locate the authorization you want to update, and select **Reauthorize**.

When reauthorizing through this page, the update is applied to all connectors and dynamic picklists that were using the authorization.

Note: Salesforce only allows 5 active access tokens at a time. If you have re-authorized the Salesforce or Salesforce Workflow connector more than 5 times, the oldest access token will be revoked. Select an existing Salesforce authorization from the Authorize dropdown to use an existing access token.

Refer to our Salesforce OAuth Token Limit article for more information and troubleshooting details.

Salesforce Workflow Connector Actions

To begin configuring the Salesforce Workflow Connector, select an **Action** and **Record Type**.

Use the list below to jump to instructions for each supported action.

Supported Actions:

- **Create a Record**: create a new record in Salesforce
- **Update a Record**: locate and update an existing record in Salesforce
- **Lookup a Record**: locate an existing record in Salesforce
- **Upload a File**: add a file to a Salesforce record

Supported Record Types

- All standard and custom record types
- **Upload a File** supports the Attachment and ContentVersion fields
- To show newly added Objects/Fields, click **Refresh Lists** located at the bottom of the action cards.

More Uses:

- Actions may be used to handle repeatable fields and values. Learn more about how to configure repeated values in the **Repeated Content** section of this document.
- Salesforce data can be passed to forms or stored in variables to use elsewhere in the workflow. Learn more about how to set up this sort of configuration in the **Prefill and Variables** section of this document.
- Each time the Salesforce Workflow Connector runs, logs are published to describe the details of the connector actions. Learn how to access logs in the **Connector Logs** section.

Preconditions and Dependencies

Use the Configuration tab to create the Actions your connector will complete when it runs. Each Salesforce Workflow Connector can support multiple actions with the ability to pass Salesforce data between actions. The Configuration tab is only available after completing the Authorization tab.

Refer to our Workflow-native Connectors article for general information on Actions and the Action Menu.

Preconditions

Preconditions determine whether a connector action runs or is skipped. If the logic set in this section resolves to true, the action will run. If the logic is false, the action will not run.

Add a Precondition to an Action

- Select **Add Precondition** from the Action Menu
- Select a **form field**, enter a **value or formula**, or select a **variable** value
    - The action will only run if the selected or entered condition is True.
    - If multiple conditions are added, you may specify whether you want the action to run if **all conditions** are satisfied, **any condition** is satisfied, or by using **custom** boolean logic. For more information about custom logic, refer to the Set Condition section of our Conditional Ruleset article .

For those familiar with the form-native Salesforce Connector, preconditions replace the function of skip-if formulas, but they work inversely.

In the form-level Salesforce Connector:

- If a skip-if formula returns TRUE, the connector step is skipped.
- If the skip-if formula returns FALSE, the connector step executes.

With Preconditions in the Salesforce Workflow Connector:

- If **no** preconditions are added, the action will **always execute**.
- If preconditions are added, the action will only execute if the defined preconditions are met.

Dependencies

Action Dependencies allow you to control the order in which actions run by making one action dependent on the execution of another. A dependent action will only run if at least one of its prerequisite actions has been executed. If all prerequisite actions are skipped, the dependent action is also skipped.

To add a dependency to an action:

- Select **Set Dependencies**
  - **Note**: This option is only available if the connector contains two or more actions.
- In the "Run after" field, select the action(s) that are the prerequisites for this action to run.
  - **Note**: If more than one dependency is added to the "Run after" field, the action will run if any of the listed prerequisite actions are run.

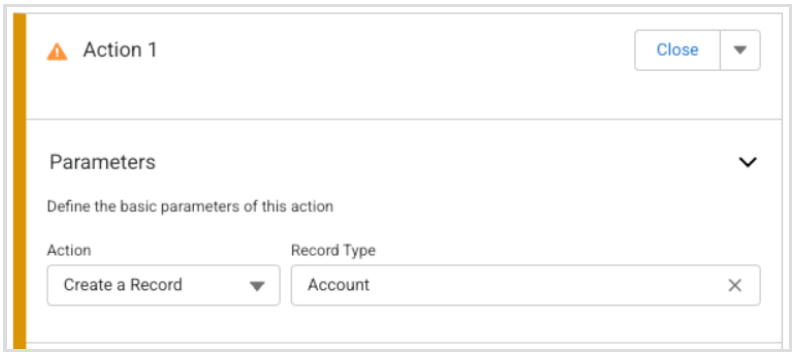
Dependencies do not determine whether an action should run in the first place—that is handled by preconditions—but they ensure that certain actions only execute after others. This is useful when actions need to follow a specific sequence to maintain logical processing or data consistency.

Actions with preconditions can be linked to dependent actions by setting action dependencies. Dependent actions run if the prerequisite action runs, and dependent actions are skipped if the prerequisite actions are skipped. In the Salesforce Workflow Connector, when one action needs to wait for another, such as creating a parent record before a related child record, dependencies can help maintain the correct sequence. By adding dependencies, you ensure that actions run in a structured and predictable way, giving you more control over the flow of your process.

Create a Record

Use the **Create a Record** action to create a new record in your Salesforce account.

Parameters

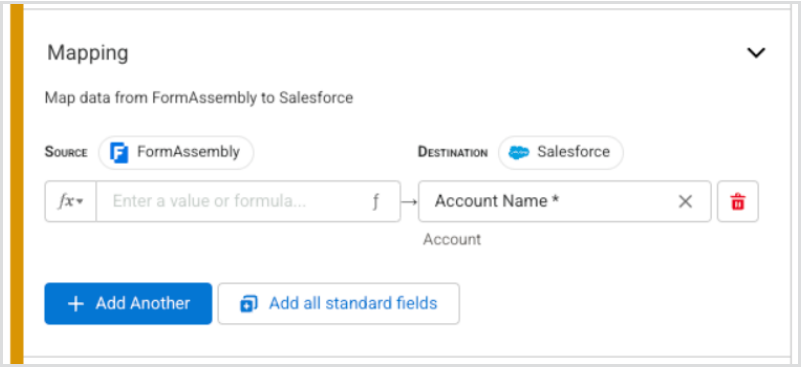


- To create a record in Salesforce, start by selecting **Create a Record** in the **Action** field within the Parameters section
- Select the type of record you want to create in the **Record Type** field
- When both Action and Record Type fields have selections, the **Mapping**, **Return Variables**, and **Additional Options** sections become available within the action card

Mapping

Define which Salesforce fields should receive FormAssembly data for the selected record type when the record is

created.



- **Add Sources to Map**
  - **Required Fields** for the selected Record Type will be added to the mapping section by default
    - Required fields for the selected Record Type are indicated with an asterisk (*) in the **Destination** dropdown
  - Optional: Select **Add all standard fields** to quickly map all standard form fields in the workflow's forms.
  - Optional: Add additional Salesforce fields to be mapped with form data by selecting the **Add Another** button
    - In the **Source** field, select the FormAssembly data to be added to the selected Salesforce field
      - **Form Field**: data entered into a form field by a respondent
      - **Value or Formula**: static text, aliases, formulas, or calculations
      - **Variable**: variables created in this connector or other Workflow-Native Connectors of this workflow
- **Map Sources to Destinations**
  - In the **Destination** field, select the Salesforce record's field to receive FormAssembly data
- **Delete** mapped rows by selecting the red trashcan icon

Return Variables

Optionally capture the ID or URL of the newly created Salesforce record to be used in other Salesforce Workflow Connector actions or Workflow-native Connectors within the same workflow.

- In the **Result From** field, select ID or URL
  - **ID**: the ID number of the newly created record
  - **URL**: the URL of the newly created record
- In the **Destination** field, open the **Value** dropdown to select Form Field or Variable
  - **Form Field**: assign the ID or URL value to a field in a workflow form
  - **Variable**: save the ID or URL as a custom variable to be used later in a Salesforce Workflow Connector action step or other Workflow-native Connectors within the same workflow
- Select **Add Variable** to add additional rows to save both ID and URL, or save them with different Destination options

Note: The Return Variable ID is similar to the SFA_OBJECT alias in the form-level Salesforce connector.

Additional Options

Configure additional options on the Create a Record action

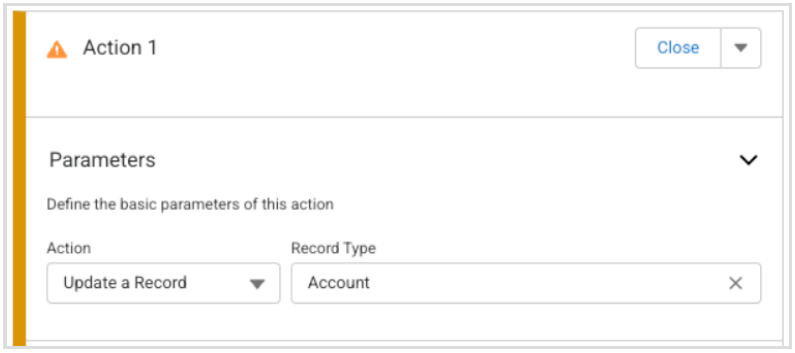
- Check the option "**Assign using active assignment rules"** to leverage existing assignment rules in Salesforce after the new record is created

---

Update a Record

Use the Update a Record action to locate a record in your Salesforce account and update the record's fields.

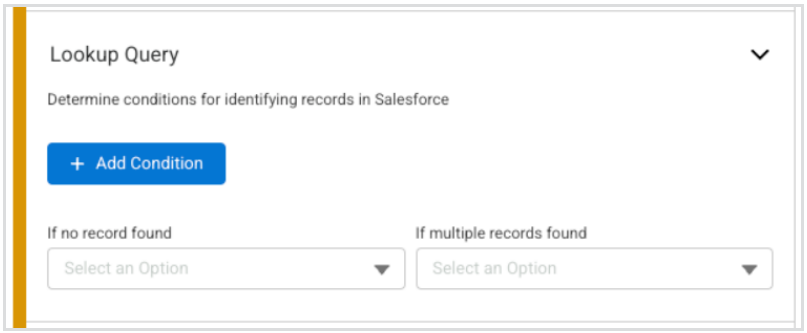
Parameters



- To update a record in Salesforce, start by selecting **Update a Record** in the **Action** field within the Parameters section
- Select the type of record you want to create in the **Record Type** field
- When both Action and Record Type fields have selections, the **Lookup Query**, **Mapping**, **Return Variables**, and **Additional Options** sections become available within the action card

Lookup Query

Set conditions to identify a matching record in Salesforce.



- Click **Add Condition** to create the first Lookup Query row
  - In the left-hand dropdown, select the **Salesforce field** to inspect for a matching value
  - In the middle dropdown, select the **logical operator** used to compare the Salesforce field to the query

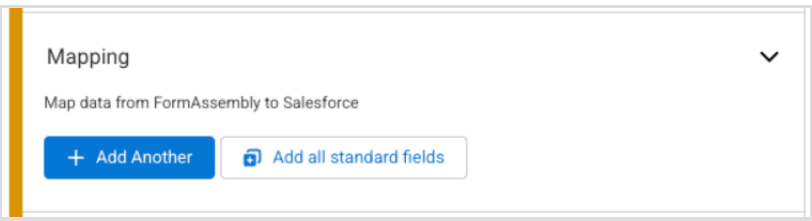
value
    - In the right-hand dropdown, select the **query value** to be used to find a matching record.
        - Select Form field, Value or Formula, or Variable to select the value type.
            - **Form Field**: data entered into a form field by a respondent
            - **Value or Formula**: static text, aliases, formulas, or calculations
            - **Variable**: variables created in this connector or other Workflow-Native Connectors of this workflow
- Optional: Add additional search criteria by selecting the **Add Condition** button
    - If more than one condition is present, select an option:
        - **All conditions must be satisfied**: all conditions entered in the Lookup Query section must be true in the matching record
        - **Any condition must be satisfied**: Only one condition entered in the Lookup Query section must be true in the matching record
        - **Custom**: Use boolean logic to customize the logical operation
- In the field, "**If no records found**", select what the connector should do if a match is not found using the above conditions
    - **Skip this action**: the remainder of the action is skipped, and the next action or workflow step is started
    - **End execution and return an error**: the connector stops all actions, writes an error to the connector log, and error handling preferences are employed
    - **Continue with dependent actions**: the connector continues with dependent actions as if this action had been completed

> **Note**: Different from the form-level Salesforce Connector, you won't see an option to create a new record if no match is found. To accomplish this, you'll need to select "**Continue with dependent actions**", return a **Variable**, and create a separate action using that variable. This allows update record actions and new record mappings to be treated separately from one another. See the example here.

- In the field, "**If multiple records found**", select what the connector should do if multiple matches are found using the above conditions
    - **Skip this action**: the remainder of the action is skipped, and the next action or workflow step is started
    - **End execution and return an error**: the connector stops all actions, writes an error to the connector log, and error handling preferences are employed
    - **Pick most recently modified record**: the record most recently modified among the matching results is selected for the record update action

Mapping

Define which Salesforce fields of the identified record should be updated



- **Add Sources to Map**
    - Optional: Select **Add all standard fields** to quickly map all standard form fields in the workflow's forms.
    - Add Salesforce fields to be mapped with form data by selecting the **Add Another** button
        - In the **Source** field, select the FormAssembly data to be added to the selected Salesforce field

        - **Form Field**: data entered into a form field by a respondent
        - **Value or Formula**: static text, aliases, formulas, or calculations
        - **Variable**: variables created in this connector or other Workflow-Native Connectors of this workflow
- **Map Sources to Destinations**
    - In the **Destination** field, select the Salesforce record's field to receive FormAssembly data
- **Delete** mapped rows by selecting the red trashcan icon

Return Variables

Optionally capture the ID or URL of the newly created Salesforce record to be used in other Salesforce Workflow Connector actions or Workflow-native Connectors within the same workflow

- In the **Result From** field, select ID or URL
    - **ID**: the ID number of the newly created record
    - **URL**: the URL of the newly created record
- In the **Destination** field, open the **Value** dropdown to select Form Field or Variable
    - **Form Field**: assign the ID or URL value to a field in a workflow form
    - **Variable**: save the ID or URL as a custom variable to be used later in a Salesforce Workflow Connector action step or other Workflow-native Connectors within the same workflow
- Select **Add Variable** to add additional rows to save both ID and URL, or save them with different Destination options

Additional Options

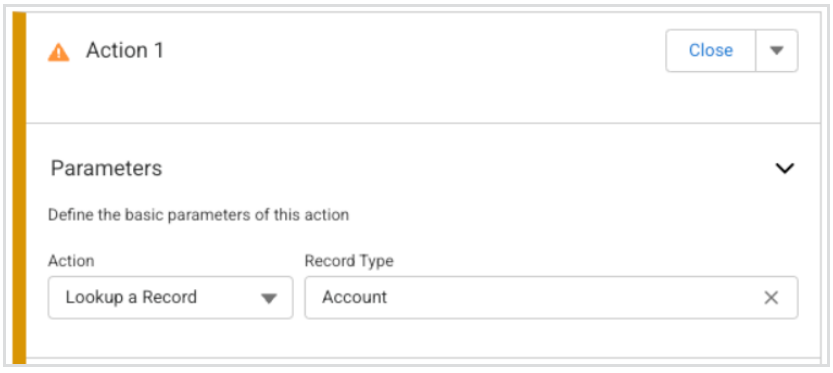
Configure additional options on the Update a Record action

- Check the option "**Prevent empty values from overwriting existing data**" to preserve the existing data in the Salesforce field if the mapped value is empty (a form field without input text, for example)

Lookup a Record

Use the **Lookup a Record** action to look up a record in your Salesforce account and return the record's data.

Parameters

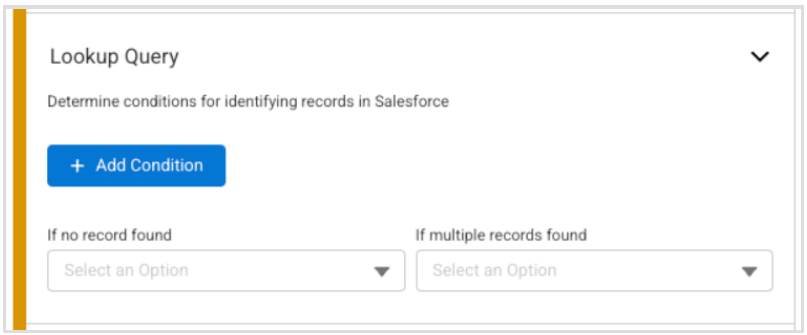


- To look up a record in Salesforce, start by selecting **Lookup a Record** in the **Action** field within the Parameters section

- Select the type of record you want to look up in the **Record Type** field
- When both the Action and Record Type fields have selections, the **Lookup Query** and **Mapping** sections become available within the action card

Lookup Query

Set conditions to identify a matching record in Salesforce



- Click **Add Condition** to create the first Lookup Query row
  - In the left-hand dropdown, select the **Salesforce field** to inspect for a matching value
  - In the middle dropdown, select the **logical operator** used to compare the Salesforce field to the query value
  - In the right-hand dropdown, select the **query value** to be used to find a matching record.
    - Select Form field, Value or Formula, or Variable to select the value type.
      - **Form Field**: data entered into a form field by a respondent
      - **Value or Formula**: static text, aliases, formulas, or calculations
      - **Variable**: variables created in this connector or other Workflow-Native Connectors of this workflow
- Optional: Add additional search criteria by selecting the **Add Condition** button
  - If more than one condition is present, select an option:
    - **All conditions must be satisfied**: all conditions entered in the Lookup Query section must be true in the matching record
    - **Any condition must be satisfied**: Only one condition entered in the Lookup Query section must be true in the matching record
    - **Custom**: Use boolean logic to customize the logical operation
- In the field, "**If no records found**", select what the connector should do if a match is not found using the above conditions
  - **Skip this action**: the remainder of the action is skipped, and the next action or workflow step is started
  - **End execution and return an error**: the connector stops all actions, writes an error to the connector log, and error handling preferences are employed
  - **Continue with dependent actions**: the connector continues with dependent actions as if this action had been completed
- In the field, "**If multiple records found**", select what the connector should do if multiple matches are found using the above conditions
  - **Skip this action**: the remainder of the action is skipped, and the next action or workflow step is started
  - **End execution and return an error**: the connector stops all actions, writes an error to the connector log, and error handling preferences are employed
  - **Pick most recently modified record**: the record most recently modified among the matching results is selected for the record update action
  - **Select all records**: select all of the Salesforce records that match the above conditions

Mapping

Capture Salesforce data from the matching record(s) to be placed in a form field (prefill) or stored as a variable

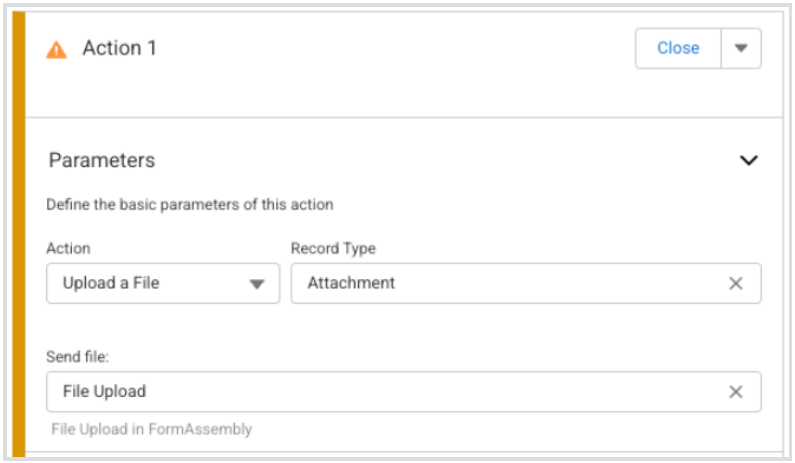
- Select the **Add Another** button to add a row to the Mapping section
- In the **Source** field, select the Salesforce field within the matched record(s) to bring into the workflow
- In the **Destination** field, select the FormAssembly data to be added to the selected Salesforce field
    - **Form Field**: prefill a field in a workflow form with the selected Salesforce data
    - **Variable**: capture the selected Salesforce data as a custom variable to be used in a Salesforce Workflow Connector action or other Workflow-native Connectors
- **Delete** mapped rows by selecting the red trashcan icon

Note: Different from the form-level Salesforce Connector, you won't see options to map to a lookup result or the ID of an object above. Instead, when you perform a lookup, you can save Salesforce fields as custom variables by clicking the selector button attached to the Destination field and changing the selection from Form Field to **Variable**. Give the variable a unique name, and you'll then be able to use it throughout the Salesforce Connector and all other Workflow-native Connectors. See the example here.

Upload a File

Use the **Upload a File** action to upload a file to a Salesforce record.

Parameters



- To upload a file to a Salesforce record, start by selecting **Upload a File** in the Action field within the Parameters section
- Select the type of record you want to create in the **Record Type** field
    - **Upload a File** supports the following record types:
        - **Attachment –** An Attachment record in Salesforce is a record of the Attachment Object, which is then associated with a parent record. Attachments appear in the Notes & Attachments Related List.
        - **Content Version –** Salesforce Files use ContentVersion records, which are then associated with a parent record. Files appear in the Files Related List on the parent record.
- When both Action and Record Type fields have selections, the **Send file** field and **Return Variables** section become available within the action card

- In the **Send file** field, select a file upload field contained within the workflow's forms
- When the file upload field has been selected in the Send file field, the **Mapping** section becomes available within the action card

Mapping

Define the destination and details of the uploaded file in Salesforce

- **Required Fields** for the selected Record Type are indicated with an asterisk (*****) in the **Destination** dropdown
  - If the **Attachment** record type is selected, **Parent ID** and **File Name** are required
    - Map the record's ID in the destination field **Parent ID** to set the parent record for the file upload
    - Map text to the destination field **File Name** to set the file name in Salesforce
  - If the **Content Version** record type is selected, **First Publish Location ID** and **Title** are required
    - Map the parent record's ID in the destination field **First Publish Location ID** to set the parent record for the file upload
    - Map text to the destination field **Title** to set the file name in Salesforce
- Add additional Salesforce fields to be mapped with form data by selecting the **Add Another** button
- In the **Source** field, select the FormAssembly data to be added to the selected Salesforce field
  - **Form Field**: data entered into a form field by a respondent
  - **Value or Formula**: static text, aliases, formulas, or calculations
  - **Variable**: variables created in this connector or other Workflow-Native Connectors of this workflow
- In the **Destination** field, select the Salesforce record's field to receive FormAssembly data
- **Delete** mapped rows by selecting the red trashcan icon

> **Note**: The Parent ID of a Salesforce record can be collected with a **Lookup a Record** action and passed to the **Upload a File** action by mapping to a variable. See the example here.

Return Variables

Optionally capture the ID or URL of the uploaded file to be used in other Salesforce Workflow Connector actions or Workflow-native Connectors within the same workflow

- In the **Result From** field, select ID or URL
  - **ID**: the ID number of the updated record
  - **URL**: the URL of the updated record
- In the **Destination** field, open the **Value** dropdown to select Form Field or Variable

    - **Form Field**: assign the ID or URL value to a field in a workflow form
    - **Variable**: save the ID or URL in a custom variable to be used later in a Salesforce Workflow Connector action step or other Workflow-native Connectors within the same workflow
- Select **Add Variable** to add additional rows to save both ID and URL, or save them with different Destination options

---

Repeated Content

When repeated fields are mapped in the Salesforce Workflow Connector, the connector can be configured to properly handle the repeated values.

The Repeated Content section becomes available on the action when a repeated field is selected.

- **"Add a record for every submitted"** field instructs the connector to repeat the action **for each repeatable section** or **once for the workflow response.**
    - Select **Workflow** to have the action run a single time, regardless of the repeated content. Values will be concatenated.
    - Select a repeatable section to have the defined action repeat as many times as the respondent repeated the section.

> **Note**: Repeatable fields not within repeatable sections will not show up as options in this dropdown. Please place these fields within repeatable sections if you wish to add a record for each submitted repeatable field.

When a repeatable field is mapped in the Salesforce Workflow Connector's Lookup Query section, the option "**Select all records"** in the field "**If multiple records found"** can be used to update all repeated values that are matched in Salesforce.

> **Note**: For more information on mapping repeatable fields, refer to the article Workflow Step - Add a Connector.

Prefill and Variables

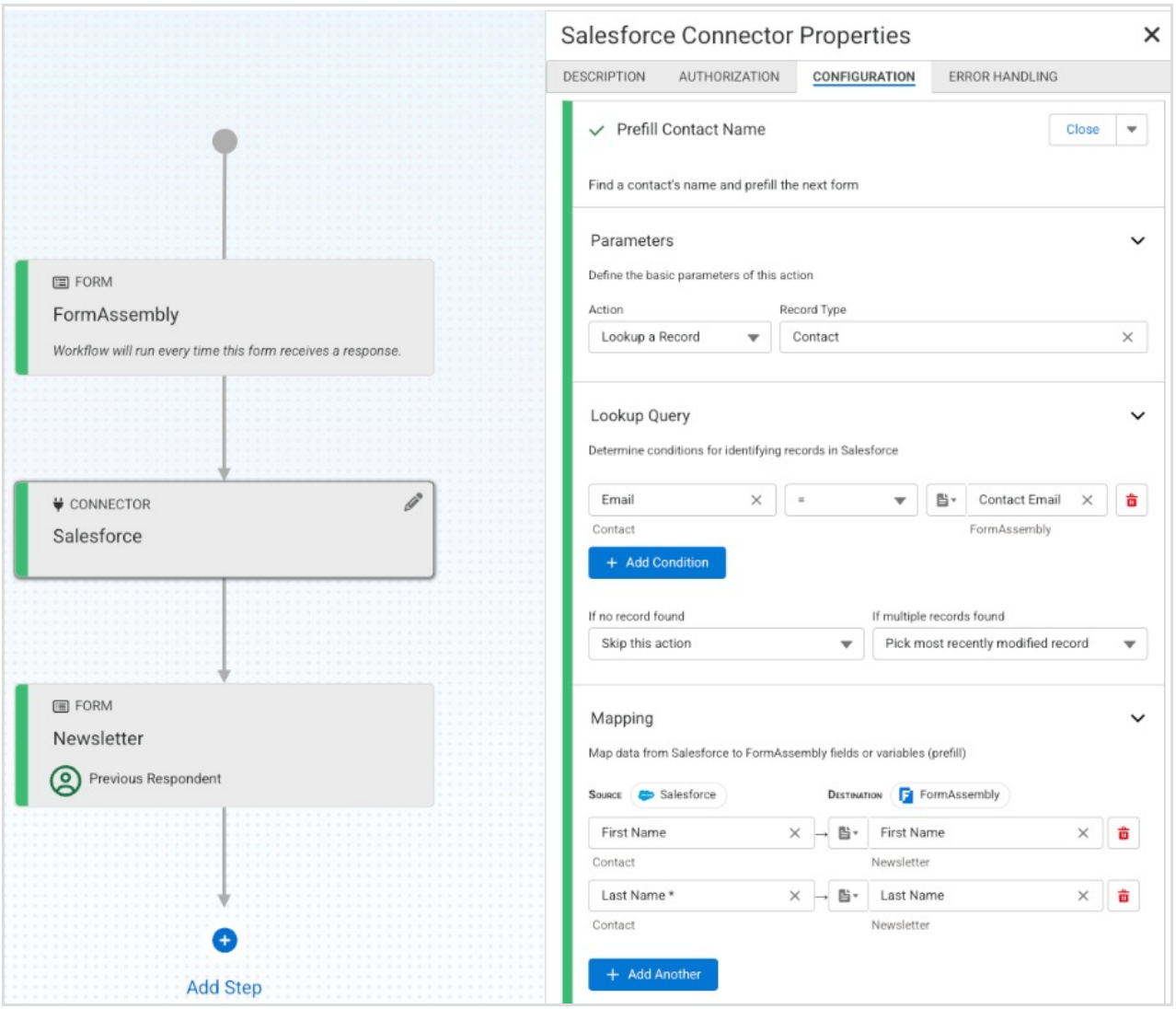
Use the **Lookup a Record** action to prefill Salesforce data in forms or store Salesforce data in variables to use elsewhere in the workflow.

Prefill

Using a **Lookup a Record** action, Salesforce record data can be retrieved and placed in a form field within the same workflow.

To prefill a form using the Salesforce Workflow Connector:

- Configure a **Lookup a Record** action to locate a desired record
- In the Lookup a Record action's **Mapping** section,
    - Select the Salesforce data you wish to retrieve from the matching record in the **Source** field
    - Select the form field you wish to prefill in the **Destination** field



The **Lookup a Record** action, combined with **URL Parameters**, can be leveraged to perform a **First-Form Prefill**. For more information about how to set this up, please reference this example.

Variables

Salesforce data can be stored in variables and passed to additional Salesforce Workflow Connector steps and other Workflow-native Connectors.

To store Salesforce data as a variable and use it in the workflow:

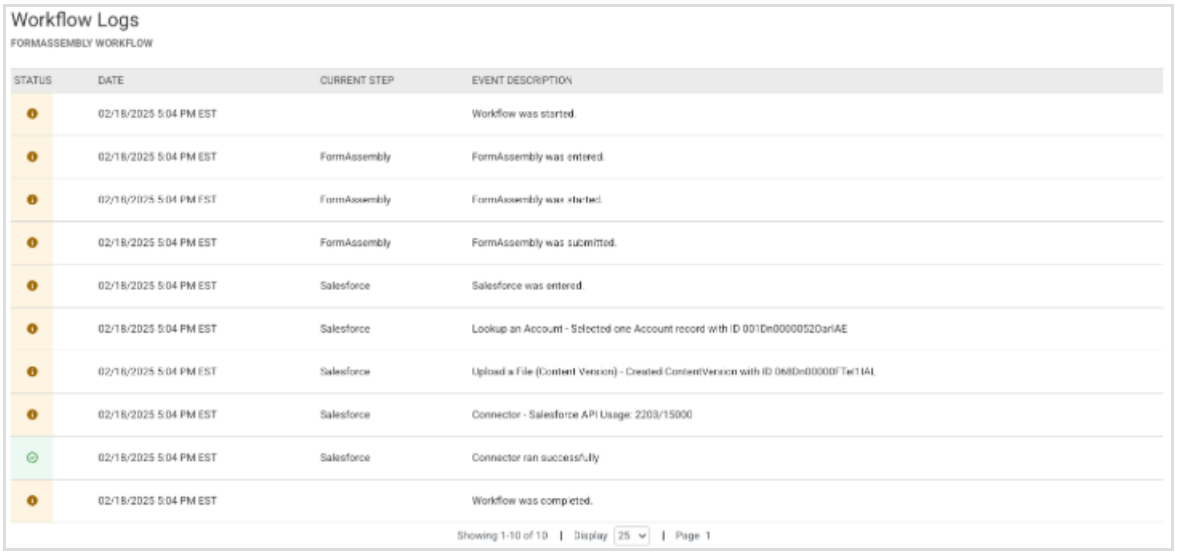
- Configure a **Lookup a Record** action to locate a desired record
- In the Lookup a Record action's **Mapping** section,
    - Select the Salesforce data you wish to retrieve from the matching record in the **Source** field
    - In the **Destination** field, select **Variable** from the value type dropdown
    - Enter a custom variable name

See an example of this setup used for a file upload here .

> **Note**: Variables that share the same name will be overwritten.

Connector Logs

Each time the Salesforce Workflow Connector runs, logs are published to describe the details of the connector actions.



To review Salesforce Workflow Connector logs:

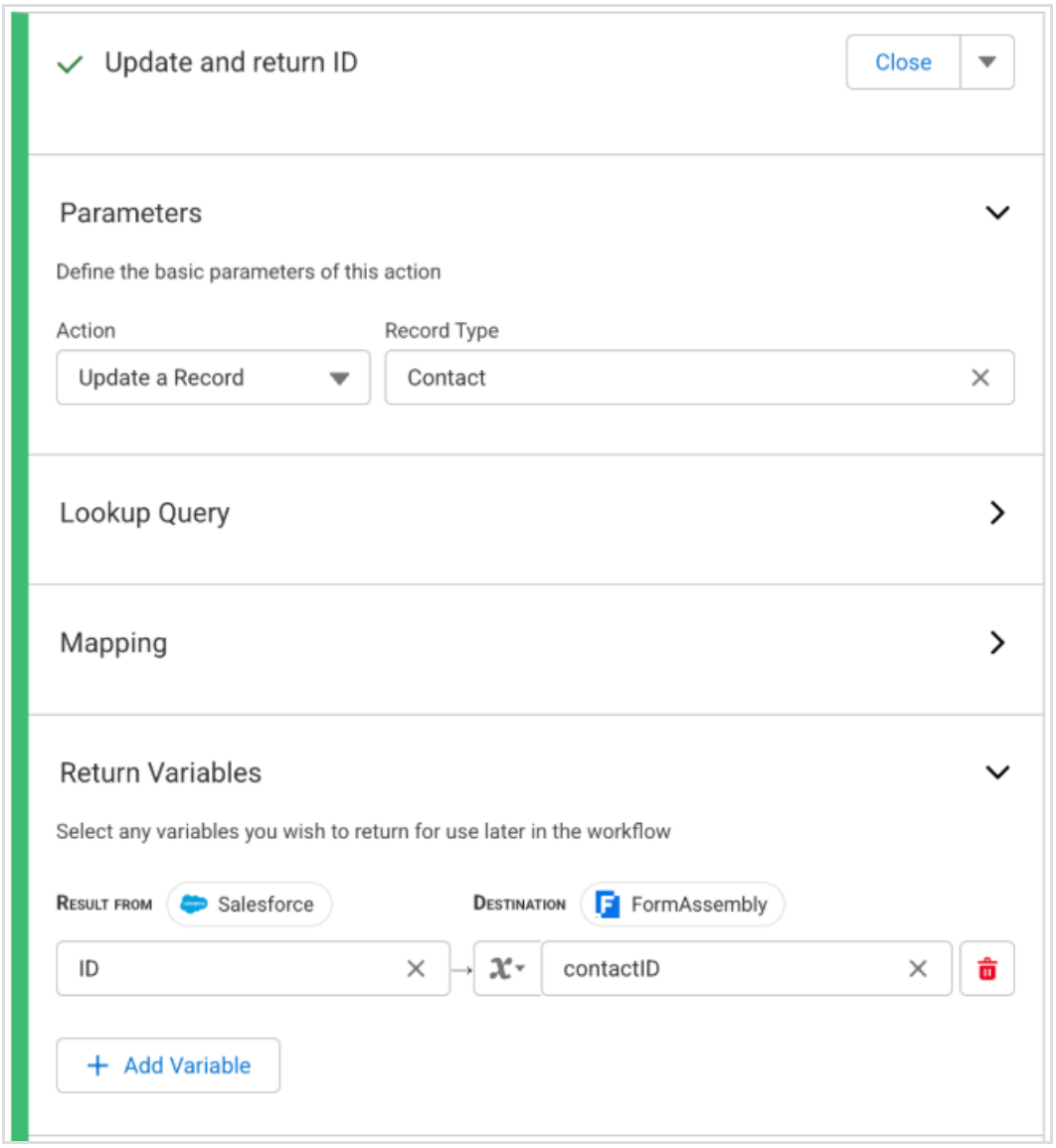
- While in the workflow builder, click on the **Responses** link on the left-hand navigation menu
- Select a response and click **View** under the Actions column
- In the upper right-hand corner of the response details page, click **View Logs**

> **Note**: Workflow Logs will show successful actions as well as errors. This is a useful tool for troubleshooting connector configuration issues.

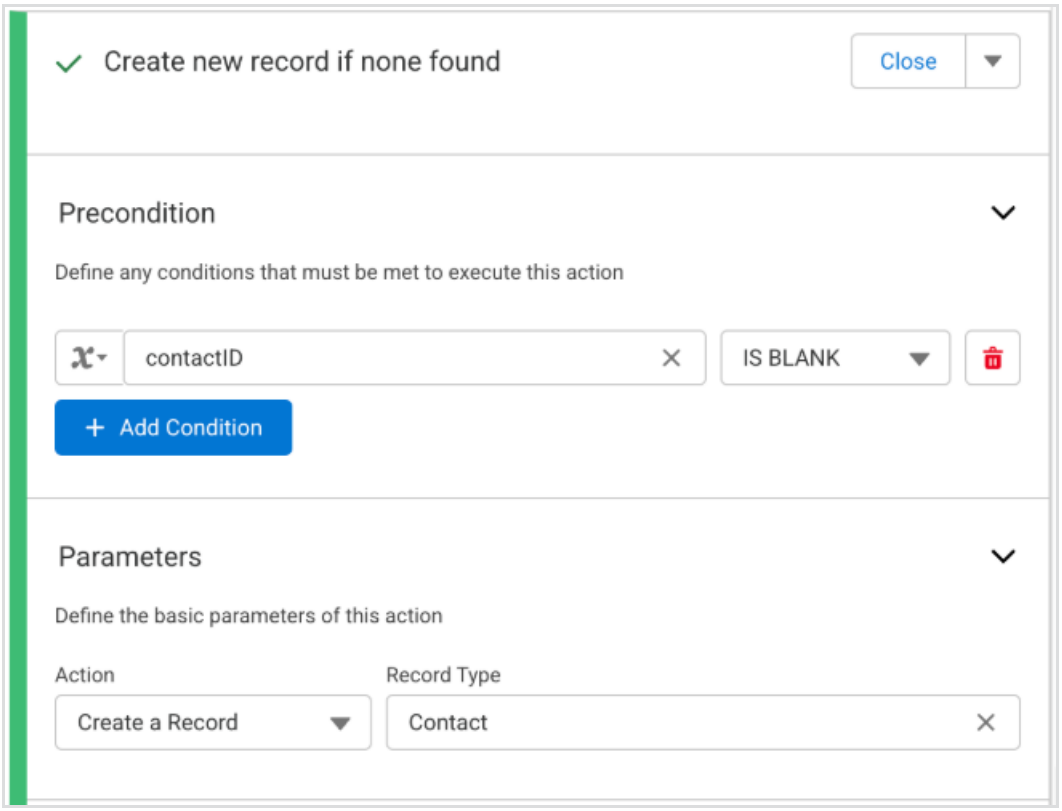
Examples

Update a record and create a new record if no match is found

Step 1: Configure the update record action and return the record ID as a variable.

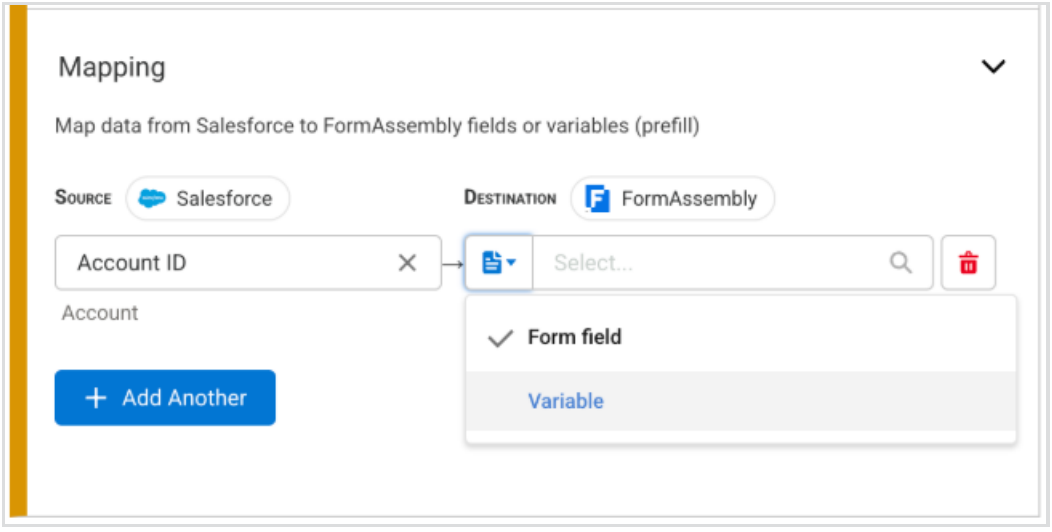


Step 2: Configure a subsequent create record action using the variable as a precondition. If the variable is blank, then the record doesn't exist, and the precondition will be met. The create record action will then execute.

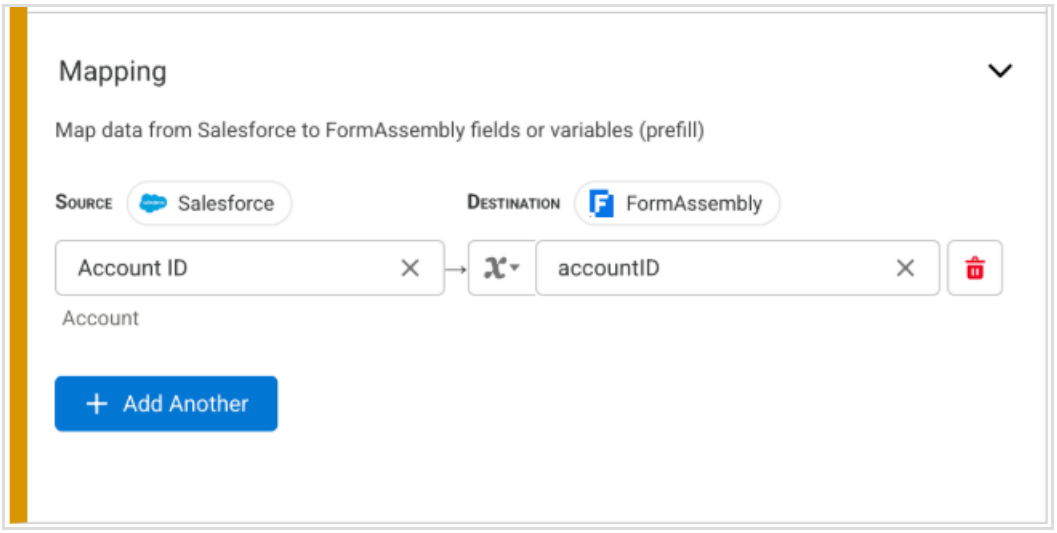


Save Salesforce fields as custom variables to use throughout Workflow-native Connectors

Step 1: Map a Salesforce field as a Variable.

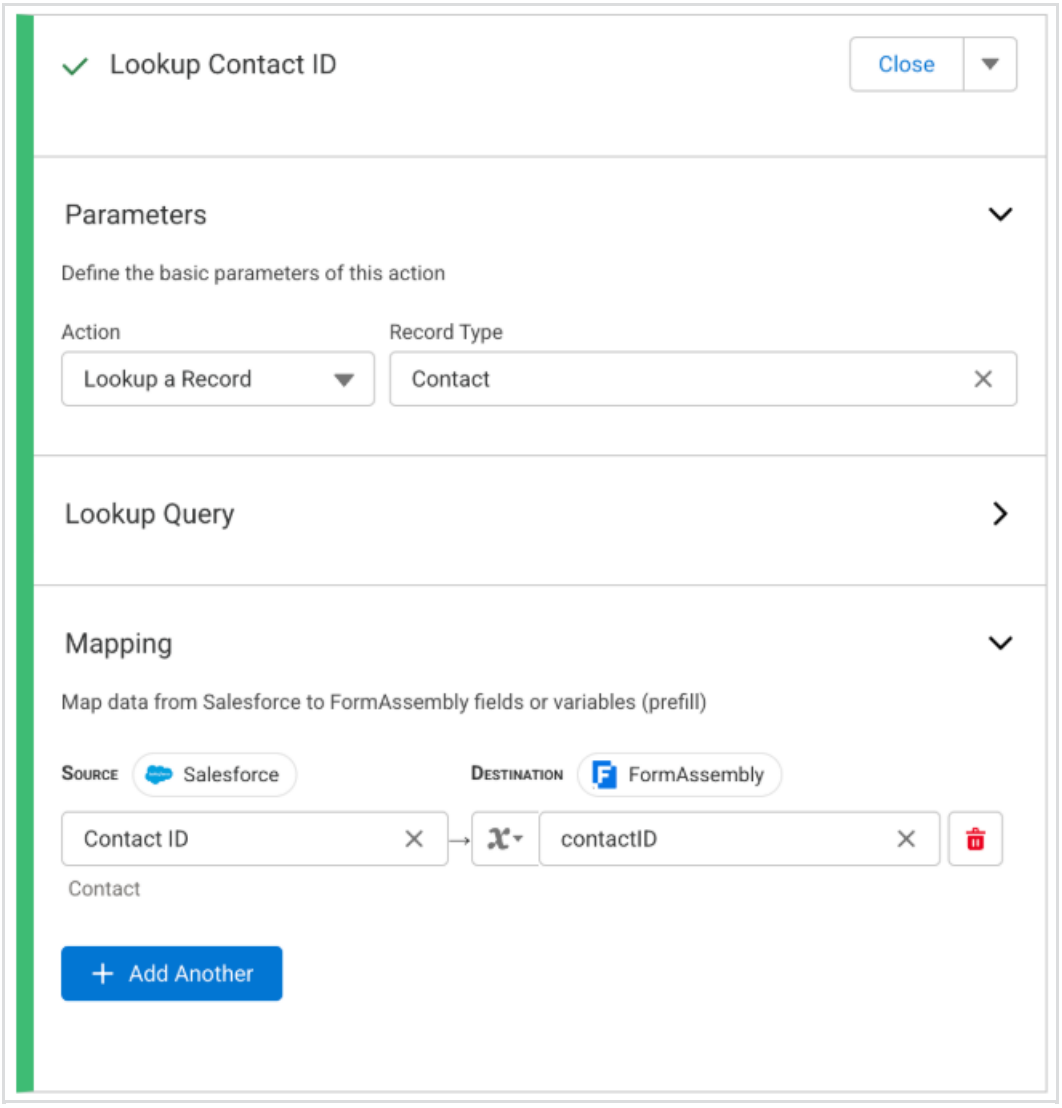


Step 2: Name the variable something unique to be used across your workflow with other Workflow-native Connectors.

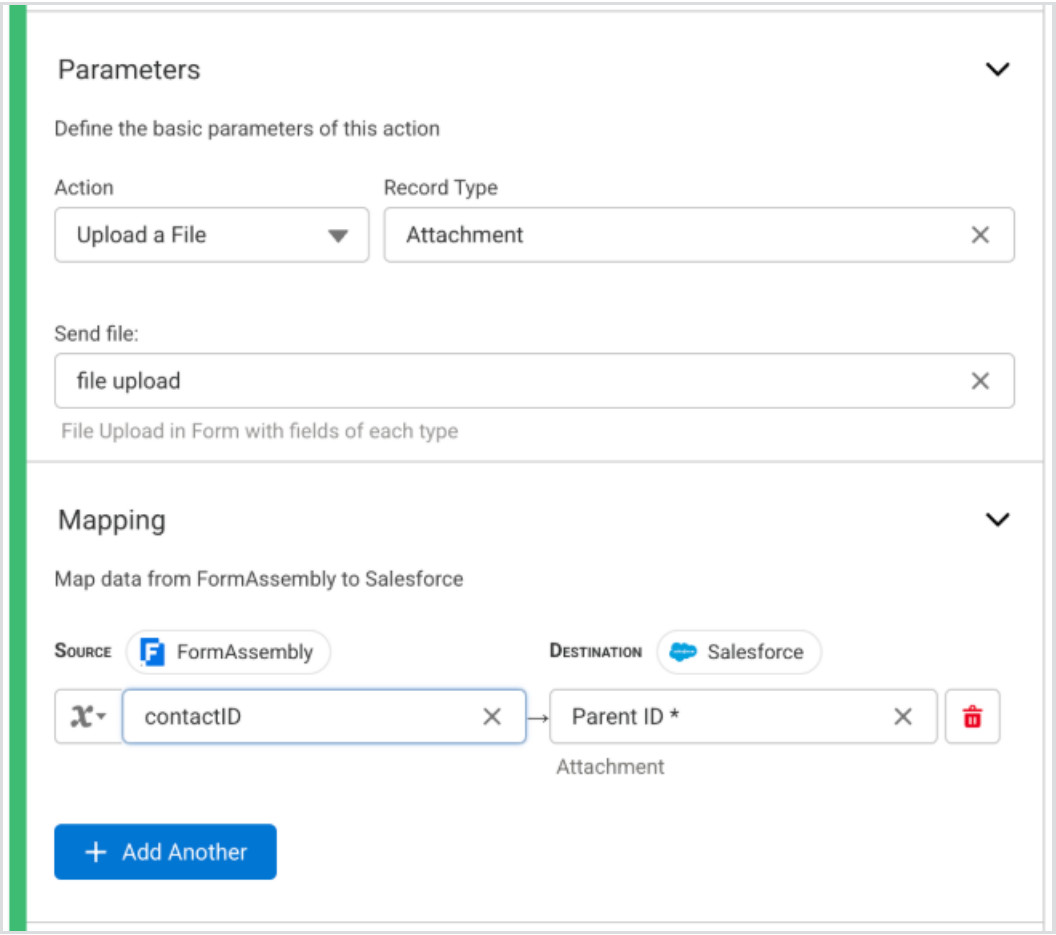


Collect the Parent ID of a Salesforce record and pass it to an Upload a File action

Step 1: Configure one action to perform a lookup and save the ID as a variable in the lookup mapping section.



Step 2: Configure a subsequent action to upload a file, and map the ID variable to the destination parent ID.



First-Form Prefill using URL Parameters

You can prefill the first form in your Workflow using data from Salesforce by adding a Salesforce Workflow Connector step before your Form Step, even as the first step in your workflow.

Step 1: Open the **Workflow Properties** panel and create a **URL Parameter**.

Step 2: Add a **Salesforce Workflow Connector** step at the beginning of your Workflow.

Step 3: Set the connector Action to **Lookup a Record**, and select the Salesforce object you want to use for prefilling.

Step 4: In the **Lookup Query** section, click **Add Condition**.

- Select a Salesforce field (e.g., *Contact ID*) and map it to a **Workflow Variable** by clicking the variable icon and selecting the variable you created in Step 1.

Step 5: Continue setting up the Lookup Query as needed, including handling multiple or missing records.

Step 6: Map fields from the retrieved Salesforce record into your first **Form Step**.