

Flow Example Setup Instructions for

Dynamic Selector



Prerequisites

- Salesforce Setup:
- In the backend of Salesforce you will need to go to Security -> Trusted URLs and add your domain's URL with ".my.salesforce.com" overriding ".lightning.force.com". For example my sample URL is:
 - <https://my-sample-dev-ed.lightning.force.com>
- In Salesforce I need to make a New Trusted URL, so these are settings that need to be configured:
 - API Name: This can be your naming convention
 - URL: (usin above example) <https://my-sample-dev-ed.my.salesforce.com>
 - Active: Checked
 - connect-src (scripts) : Checked
 - font-src (fonts) : Checked
 - frame-src (iframe content) : Checked
 - img-src (images) : Checked
 - media-src (audio and video) : Checked
 - style-src (stylesheets) : Checked
- Once configured it can be saved

trusted

- security
- Trusted URLs
- Trusted URLs for Redirects


didn't find what you're looking for?
y using Global Search.



SETUP Trusted URLs

Trusted URL Details

Trusted URL Detail

Trusted URL Information	
	API Name: MyOrg
	Description:
	Created By: Jeremy Hostetter , 19/03/

Content Security Policy (CSP) Settings

To help prevent cross-site scripting (XSS) and other code injection attacks, the Lightn

CSP Context	All

CSP Directives

Lightning components, third-party APIs, and WebSocket connections can load the se

connect-src (scripts)	<input checked="" type="checkbox"/>
font-src (fonts)	<input checked="" type="checkbox"/>
frame-src (iframe content)	<input checked="" type="checkbox"/>
img-src (images)	<input checked="" type="checkbox"/>
media-src (audio and video)	<input checked="" type="checkbox"/>
style-src (stylesheets)	<input checked="" type="checkbox"/>

Permissions Policy Directives

To control access to these browser features, enable the Permissions-Policy HTTP he

camera	<input type="checkbox"/>
--------	--------------------------

Creating the Flow – Variables



Below are a list of variables (be cautious of capitalization) that may need to be configured. Not all are mandatory, just the ones needed for your requirements

- API Name: input
 - Resource type: Variable
 - Data type: Apex-Defined
 - Apex Class: cadmus_core__ComponentDataByFlowInput
 - Available for input: true
 - Used for recovering record ID or object name
- API Name: docConfigs
 - Resource type: Variable
 - Allow multiple values (collection): true
 - Data type: Text
 - Available for output: true
 - Used for storing the Doc Configs that will be required for generating
- API Name: optionalDocConfigs
 - Resource type: Variable
 - Allow multiple values (collection): true
 - Data type: Text
 - Available for output: true
 - Used for storing the Doc Configs that will be optional for generating
- API Name: docxContentVersions
 - Resource type: Variable
 - Allow multiple values (collection): true
 - Data type: Text
 - Available for output: true
 - Used for selecting DOCX files that are stored in Salesforce Files

Continued.....

A screenshot of the 'New Resource' configuration window in Salesforce Flow Builder. The window title is 'New Resource'. It contains several fields and checkboxes:

- * Resource Type:** A dropdown menu with 'Variable' selected.
- * API Name:** A text input field containing 'input'.
- Description:** An empty text area.
- * Data Type:** A dropdown menu with 'Apex-Defined' selected. To its right is a checkbox labeled 'Allow multiple values (collection)' which is unchecked.
- * Apex Class:** A text input field containing 'cadmus_core__ComponentDataByFlowInput'.
- Availability Outside the Flow:** A section with two checkboxes: 'Available for input' (checked) and 'Available for output' (unchecked).

At the bottom right, there are 'Cancel' and 'Done' buttons.

Creating the Flow – Variables



Continued.....

- API Name: pdfContentVersions
 - Resource type: Variable
 - Allow multiple values (collection): true
 - Data type: Text
 - Available for output: true
 - Used for selecting PDF files that are stored in Salesforce Files
- API Name: alternative
 - Resource type: Variable
 - Data type: Text
 - Available for output: true
 - Used if you are using alternatives in your doc config
- API Name: locale
 - Resource type: Variable
 - Data type: Text
 - Available for output: true
 - Used if you require locale formatting on your generation
- API Name: packs
 - Resource type: Variable
 - Allow multiple values (collection): true
 - Data type: Text
 - Available for output: true
 - Used if you are using packs, such as emailing your generated document

Creating the Flow – Sample Flow Creation



The following is just a sample flow. In this instance we will be putting our component on an Opportunity record.

- First create an autolaunched flow (no trigger)

- Start by adding a Get Records component to your flow.

These are the required settings:

- Object: Opportunity
- Condition Requirements: All Conditions Are Met (AND)
- Field: Id
- Operator: Equals
- Value: input>recordId (input variable configuration referenced above)

- How many records to store: Only the first record

Edit Get Records
Get Oppty Record (Get_Oppty_Record)

Get Records of This Object

*Object

Opportunity

Filter Opportunity Records

Condition Requirements

All Conditions Are Met (AND)

Field

Id

Operator

Equals

Value

input > recordId

Add Condition

Sort Opportunity Records

Sort Order

Not Sorted

If you store only the first record, filter by a unique field, such as ID.

How Many Records to Store

Only the first record

All records

How to Store Record Data

Automatically store all fields

Choose fields and let Salesforce do the rest

Choose fields and assign variables (advanced)

Creating the Flow – Sample Flow Creation

Next add another Get Records component. This will be used to get the mandatory doc configs to generate:

- Object: Doc Config
 - Condition Requirements: Based on what doc configs you will need to pull in
-
- How Many Records to Store: All records



Edit Get Records
Get No Record Test Doc Config (*Get_No_Record_Test_Doc_Config*)

Get Records of This Object

* Object
Doc Config

Filter Doc Config Records

Condition Requirements
All Conditions Are Met (AND)

Field	Operator	Value	
Name	Contains	Quick Access	
AND cadmus_core__RecordTypeName__	Equals	Main Word Document	
AND Starting_Object__c	Equals	Opportunity	

+ Add Condition

Sort Doc Config Records

Sort Order
Not Sorted If you store only the first record, filter by a unique field, such as ID.

How Many Records to Store

Only the first record

All records

How to Store Record Data

Automatically store all fields

Choose fields and let Salesforce do the rest

Choose fields and assign variables (advanced)

Creating the Flow – Sample Flow Creation



Next add a Loop Component to take the collection from the Get Doc Config Records component to start looping through the results:

- Collection Variable: API name of the Get Doc Config Get Records component

Edit Loop Loop 3 (Loop_3)

Start a loop path for iterating over items in a collection variable. For each iteration, the flow temporarily stores the item in the loop variable.

Select Collection Variable

* Collection Variable

{!Get_No_Record_Test_Doc_Config}

Specify Direction for Iterating Over Collection

* Direction

- First item to last item
 Last item to first item



To use the current item in other elements in the loop, use the API name of the Loop element. Example: if your flow iterates over accounts with a Loop element named "My_Account_Loop" you can reference the current item from that loop element. Just start typing "My_Account_Loop" and select "Current item from Loop My_Account_Loop".

Creating the Flow – Sample Flow Creation

Add an Assignment component inside the loop to **add** the ID's from the Loop to the Doc Configs variable.



Edit Assignment
Assign Mandatory Doc Configs (*Assign_Mandatory_Doc_Configs*)

Set Variable Values
Each variable is modified by the operator and value combination.

Variable	Operator	Value
<input type="text" value="A docConfigs X"/>	<input type="text" value="Add"/>	<input type="text" value="{!Loop_3.Id}"/>

[+ Add Assignment](#)

Creating the Flow – Sample Flow Creation

Next add another Get Records component. This will be used to get the **optional** doc configs to generate:

- Object: Doc Config
- Condition Requirements: Based on what doc configs you will need to pull in
- How Many Records to Store: All records



Edit Get Records
Get Academy Quote Doc Config (Get_Academy_Quote_Doc_Config)

Get Records of This Object

* Object
Doc Config

Filter Doc Config Records

Condition Requirements
All Conditions Are Met (AND) ▼

Field	Operator	Value	
Name	Contains ▼	Academy	
AND cadmus_core__RecordTypeName__	Equals ▼	Main Word Document	
AND Starting_Object__c	Equals ▼	Opportunity	

[+ Add Condition](#)

Sort Doc Config Records

Sort Order
Not Sorted ▼ If you store only the first record, filter by a unique field, such as ID.

How Many Records to Store

Only the first record

All records

How to Store Record Data

Automatically store all fields

Choose fields and let Salesforce do the rest



Choose fields and assign variables (advanced)

Creating the Flow – Sample Flow Creation



Next add a Loop Component to take the collection from the Get Doc Config Records component to start looping through the results:

- Collection Variable: API name of the Get Doc Config Get Records component

 Edit Loop
Loop 4 (Loop_4) 

Start a loop path for iterating over items in a collection variable. For each iteration, the flow temporarily stores the item in the loop variable.

Select Collection Variable


* Collection Variable

Specify Direction for Iterating Over Collection

* Direction

First item to last item

Last item to first item

 To use the current item in other elements in the loop, use the API name of the Loop element. Example: if your flow iterates over accounts with a Loop element named "My_Account_Loop" you can reference the current item from that loop element. Just start typing "My_Account_Loop" and select "Current Item from Loop My_Account_Loop".

Creating the Flow – Sample Flow Creation



Add an Assignment component inside the loop to add the ID's from the Loop to the Doc Configs variable.

- Variable: optionalDocConfigs (definition of variable referenced above)
 - Operator: Add
- Value: the current record ID from the loop

The screenshot shows the 'Edit Assignment' configuration window in PDF Butler. The title bar reads 'Edit Assignment' and 'Assign Optional Doc Configs (Assign_Optional_Doc_Configs)'. Below the title bar, the section is titled 'Set Variable Values' with the instruction 'Each variable is modified by the operator and value combination.' There are three input fields: 'Variable' containing 'optionalDocConfigs', 'Operator' set to 'Add', and 'Value' containing '{!Loop_4.Id}'. A '+ Add Assignment' button is located at the bottom left of the configuration area.

Edit Assignment
Assign Optional Doc Configs (Assign_Optional_Doc_Configs)

Set Variable Values
Each variable is modified by the operator and value combination.

Variable	Operator	Value
optionalDocConfigs	Add	{!Loop_4.Id}

+ Add Assignment

Creating the Flow – Sample Flow Creation



In this example we will be also getting PDF files directly related to the record we are generating from.

Add a Get Record component:

- Object: Content Document Link
 - Condition Requirements: All Conditions Are Met (AND)
 - Field: LinkedEntityId
 - Operator: Equals
 - Value: Id from the original Get Record Component that gets Opportunity Record
 - How Many Records to Store: All records
 - How to Store Record Data: Choose fields and let Salesforce do the rest
- Add the ContentDocumentId field

Edit Get Records
Get Linked Files Related To Oppty (*Get_Linked_Files_Related_To_Oppty*)

Get Records of This Object

* Object
Content Document Link

Filter Content Document Link Records

Condition Requirements
All Conditions Are Met (AND)

Field	Operator	Value
LinkedEntityId	Equals	Opportunity from Get_Oppty_...

+ Add Condition

Sort Content Document Link Records

Sort Order
Not Sorted ⚠ If you store only the first record, filter by a unique field, such as ID.

How Many Records to Store

Only the first record
 All records

How to Store Record Data

Automatically store all fields
 Choose fields and let Salesforce do the rest
 Choose fields and assign variables (advanced)

Select Content Document Link Fields to Store in Variable

Field	ID
Field	ContentDocumentId

Creating the Flow – Sample Flow Creation



Next add a Loop Component to take the collection from the Get Content Document Link Records component to start looping through the results:

- Collection Variable: API name of the Get Content Document Link Get Records component

Edit Loop

Loop 1 (Loop_1)

Start a loop path for iterating over items in a collection variable. For each iteration, the flow temporarily stores the item in the loop variable.

Select Collection Variable

*Collection Variable

Specify Direction for Iterating Over Collection

*Direction

First item to last item

Last item to first item

To use the current item in other elements in the loop, use the API name of the Loop element. Example: if your flow iterates over accounts with a Loop element named "My_Account_Loop" you can reference the current item from that loop element. Just start typing "My_Account_Loop" and select "Current Item from Loop My_Account_Loop".

Creating the Flow – Sample Flow Creation



Inside the loop add a get records element to isolate the Content Document link to the ID of the Content Document object. (It is important that these steps are only grabbing PDF's, so the DOCX files will be grabbed in another step.)

- Object: Content Document
 - Condition Requirements: All Conditions Are Met (AND)
 - Criteria 1:
 - Field: Id
 - Operator: Equals
 - Value: Current ContentDocumentId from Loop
 - Criteria 2:
 - Field: FileExtension
 - Operator: Equals
 - Value: PDF
 - How Many Records to Store: Only the first record
 - How to Store Record Data: Choose fields and let Salesforce do the rest
- Add the LatestPublishedVersionId field

Edit Get Records
Get PDF Content Document File From Linked Files
(Get_PDF_Content_Document_File_From_Linked_Files)

Get Records of This Object

* Object
Content Document

Filter Content Document Records

Condition Requirements
All Conditions Are Met (AND)

Field	Operator	Value
Id	Equals	Current Item from Loop Loop...
FileExtension	Equals	PDF

+ Add Condition

Sort Content Document Records

Sort Order
Not Sorted ⚠ If you store only the first record, filter by a unique field, such as ID.

How Many Records to Store
 Only the first record
 All records

How to Store Record Data
 Automatically store all fields
 Choose fields and let Salesforce do the rest
 Choose fields and assign variables (advanced)

Select Content Document Fields to Store in Variable

Field
ID
LatestPublishedVersionId

Creating the Flow – Sample Flow Creation



Add a Decision to pick up only the PDFs using the file extension field

Then on the positive outcome add an assignment variable inside the loop to **add** ID's from the loop to content version links:

Assignment

- Variable: pdfContentVersion (defined earlier in instructions)
- Value: LatestPublishedVersion from the Content Document ~~Get~~ ~~Record component~~ current item loop from prior step.



- For example.....
{Loopgetpdfs.ContentDocument.LatestPublishedVersionId}

Creating the Flow – Sample Flow Creation



To add DOCX files, this is done with another loop. Add a Loop Component to take the collection from the Get Content Document Link Records component to start looping through the results:

- Collection Variable: API name of the Get Content Document Link Get Records component

 **Edit Loop**
Loop2 (Loop2) 

Start a loop path for iterating over items in a collection variable. For each iteration, the flow temporarily stores the item in the loop variable.

Select Collection Variable


* Collection Variable

Specify Direction for Iterating Over Collection

* Direction

First item to last item

Last item to first item

 To use the current item in other elements in the loop, use the API name of the Loop element. Example: if your flow iterates over accounts with a Loop element named "My_Account_Loop" you can reference the current item from that loop element. Just start typing "My_Account_Loop" and select "Current Item from Loop My_Account_Loop".

Creating the Flow – Sample Flow Creation



Inside the loop add a get records element to isolate the Content Document link to the ID of the Content Document object. It is important that these steps are only grabbing DOCX files:

- Object: Content Document
 - Condition Requirements: All Conditions Are Met (AND)
 - Criteria 1:
 - Field: Id
 - Operator: Equals
 - Value: Current ContentDocumentId from Loop
 - Criteria 2:
 - Field: FileExtension
 - Operator: Equals
 - Value: DOCX
 - How Many Records to Store: Only the first record
 - How to Store Record Data: Choose fields and let Salesforce do the rest
- Add the LatestPublishedVersionId field

The screenshot shows the configuration for an 'Edit Get Records' step in a Salesforce flow. The title is 'Get DOCX Content Document File From Linked Files' with a sub-label '(Get_DOCX_Content_Document_File_From_Linked_Files)'. The configuration is divided into several sections: 1. 'Get Records of This Object': The 'Object' is set to 'Content Document'. 2. 'Filter Content Document Records': 'Condition Requirements' is set to 'All Conditions Are Met (AND)'. There are two filter criteria: the first has 'Field' as 'Id', 'Operator' as 'Equals', and 'Value' as 'Current Item from Loop Loop2...'; the second has 'Field' as 'FileExtension', 'Operator' as 'Contains', and 'Value' as 'docx'. An 'Add Condition' button is visible. 3. 'Sort Content Document Records': 'Sort Order' is set to 'Not Sorted'. A warning icon and text state: 'If you store only the first record, filter by a unique field, such as ID.' 4. 'How Many Records to Store': The radio button for 'Only the first record' is selected. 5. 'How to Store Record Data': The radio button for 'Choose fields and let Salesforce do the rest' is selected. 6. 'Select Content Document Fields to Store in Variable': Two fields are listed: 'ID' and 'LatestPublishedVersionId'. A 'Tips' icon is at the bottom right.

Creating the Flow – Sample Flow Creation

Add a Decision to pick up only the DOCXs using the file extension field

Then on the positive outcome add an assignment variable inside the loop to **add** ID's from the loop to content version links:

- Variable: docxContentVersion (defined earlier in instructions)
- Value: LatestPublishedVersion from the Content Document Get Record component current item loop from prior step.
- For example
 {!Loopgetword.ContentDocument.LatestPublishedVersionId}



Creating the Flow – Sample Flow Creation



In this example we will only be grabbing one pack, but using some of the previous similar configurations from previous setups you can grab your packs logically with flow.

For this we are just hard coding a pack taking an ID from Salesforce for one Email pack that we are using.

Grab a Get Records component:

- Object: PDF Butler Pack

Edit Get Records
Get Packs (Get_Packs)

Get Records of This Object

* Object

Filter PDF Butler Pack Records

Condition Requirements

Field	Operator	Value
<input type="text" value="Id"/>	<input type="text" value="Equals"/>	<input type="text" value="a077Q000004hooAQAQ"/>

[+ Add Condition](#)

Sort PDF Butler Pack Records

Sort Order
 If you store only the first record, filter by a unique field, such as ID.

How Many Records to Store

Only the first record
 All records

How to Store Record Data

Automatically store all fields
 Choose fields and let Salesforce do the rest
 Choose fields and assign variables (advanced)

Creating the Flow – Sample Flow Creation



Add an assignment component to get pack added to variable:

- Variable: packs (variable defined earlier in instructions)
 - Operator: Add
-
- Value: ID from Get Packs component

Edit Assignment
Assignment 6 (Assignment_6)

Set Variable Values

Each variable is modified by the operator and value combination.

Variable	Operator	Value	
<input type="text" value="A_packs X"/>	<input type="text" value="Add"/>	<input type="text" value="{!Get_Packs.Id}"/>	

Creating the Flow – Sample Flow Creation

For the flow, logic can drive alternatives and locales. Define these values in an assignment variable. The variable definitions are defined earlier in the instructions.



The screenshot shows a dialog box titled "Edit Assignment" with a subtitle "Assignment (Assignment)". The main section is "Set Variable Values" with the instruction "Each variable is modified by the operator and value combination." Below this, there are two rows of configuration:

Variable	Operator	Value
A _a alternative X	Equals	Opportunity from Get_Oppty_Rec... X
A _a locale X	Equals	A _a Opportunity from Get_Oppty_Rec... X

At the bottom of the dialog is a button labeled "+ Add Assignment".

Record Page Setup



- To complete configuration, go to Salesforce Page Layout of a record from the object you are generating the document from.
- Find the component on left side of page, can filter by pdfButlerDynamicSelector. Drag it to where you want placed on page layout
- On the component configuration, insert the API name of the flow created

The screenshot shows the Salesforce Lightning App Builder interface for an Opportunity Record Page. The top navigation bar includes 'Lightning App Builder', 'Pages', and 'Opportunity Record Page'. The main workspace is divided into three sections: Components, Fields, and a central preview area. In the Components panel, a search filter 'dyna' is applied, and the 'pdfButlerDynamicSelector' component is highlighted under the 'Custom - Managed (1)' category. A red callout box points to this component with the text 'Name of the component'. The central preview area shows a sample record page with a red callout box pointing to the component's placement with the text 'Drag to the page'. The right-hand configuration panel for 'pdfButlerDynamicSelector' is visible, showing a 'Name of the flow' field with the value 'DYNAMIC_SELECTOR_DEMO' and a 'Set Component Visibility' section with a 'Filters' area containing an 'Add Filter' button. A red callout box points to the 'Filters' area with the text 'Insert flow name here'.