

1. [Lightning Flow](#)



2. [Combine the Power of Process Builder and Flow Builder](#)

Combine the Power of Process Builder and Flow Builder

Learning Objectives

After completing this unit, you'll be able to:

- Describe a business process that can be automated using a process and a flow.
- Define what a flow variable is.
- Build a flow that iterates over a group of records.
- Build a process that starts a flow.

When Process Builder Isn't Enough

Process Builder isn't designed to address every possible use case, so you may find that it can automate parts of your business process, but not all. For example, Process Builder can't:

- Post to a community feed.
- Submit a related record for approval.
- Delete records.
- Create a bunch of records and associate them with each other.
- Perform complex logic.

But there's good news! You don't have to rebuild the whole automation in another tool. Configure the more complex functionality in a flow, and then add a flow action to your process. If a flow can't do what you need, you or a developer can write Apex to do so. Then add an Apex action to your process.

Clone Records with Children

You've been asked to automatically create renewal opportunities when an opportunity is Closed Won. The renewal should be a clone of the original opportunity. We can clone records in Process Builder, but we also need to clone the products and associate them with the renewal opportunity.

In Process Builder, you can't grab the ID of the created record and use it elsewhere. Luckily, you *can* do that in a flow. Just build a flow that clones the opportunity and its products, and build a process that calls the flow when an opportunity is closed won.

Beyond the Basics

In the last unit, we talked about flows that guide users through a business process with screens. We call those *screen flows*. You can also build *autolaunched flows*, which run in the background like a process. The main difference is that autolaunched flows can't have screens, which require user interaction. Because they have no screens, you can call autolaunched flows from backend things like processes and Apex classes.

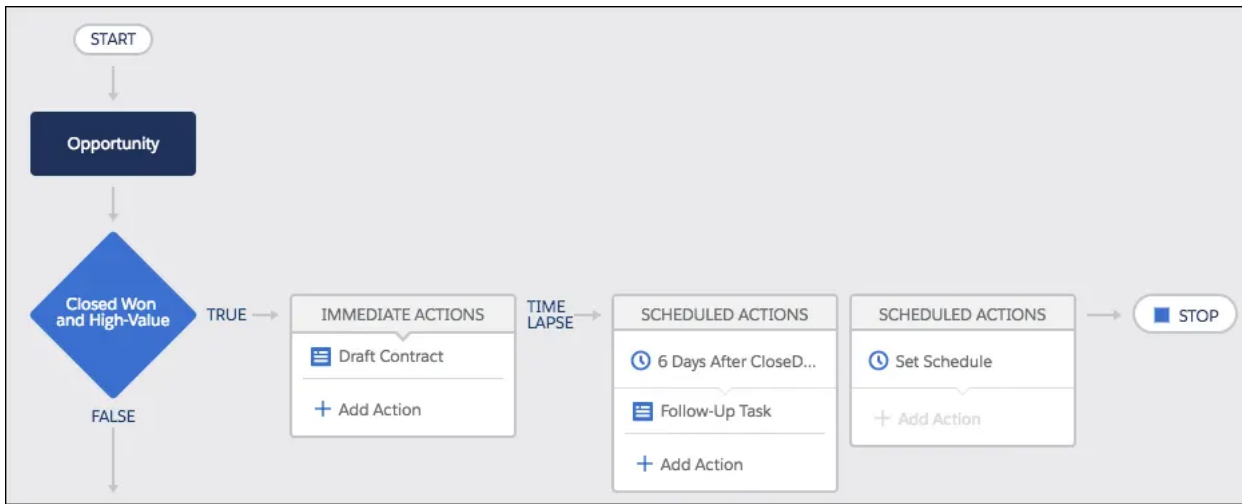
Build a Process

Instead of creating another process, let's build on the one we created in an earlier unit.



Note

You can't edit an active process. If you've already activated your process, click **Clone** to create a new version of the current process.



In the Closed Won Opportunities process, add a criteria node named Closed Won with this filter condition.

Field	Value
Opportunity > Stage	Closed Won

We can't configure the actions in Process Builder, so let's save and switch to Flow Builder.

Build a Flow

From Setup, enter **Flows** in the Quick Find box, then select **Flows**, click **New Flow**, select Autolaunched Flow, and click **Create**.

We're building a flow that:

- Clones an opportunity.
- Clones opportunity products.
- Associates the cloned products with the cloned opportunity.

To do all that, the flow needs some data from the process: the original opportunity and its opportunity products.

Flow variables come in four types.

Type	Can Store...	Example
Variable	A single value	"Hello World", true, 6
Collection Variable	Multiple values of the same data type	[1, 2, 3, 5, 8, 13]
Record Variable	A set of field values for a single record	Rating, ID, and Name for an account
Record Collection Variable	A set of field values for multiple records of the same object type	Rating, ID, and Name for multiple accounts

To store the opportunity and its opportunity products, we create a record variable and a record collection variable. Later, when we add the flow as an action in the process, we pass data into these variables, so they must allow input.

1. From the Manager in Toolbox, click **New Resource** and set these values.

Field	Value
Resource Type	Variable
API Name	opportunity
Data Type	Record

Field	Value
Object	Opportunity
Available for input	Selected

2. Click **Done**.
3. Click **New Resource** and set these values.

Field	Value
Resource Type	Variable
API Name	oppProducts_Original
Data Type	Record
Allow multiple values (collection)	Selected
Object	Opportunity Product
Available for input	Selected

4. Click **Done**.

Clone the Opportunity

When the flow is called by the process, the `{!opportunity}` variable contains fields from the original opportunity. Before we clone that opportunity, let's update some of the fields. That way, the renewal opportunity doesn't start off as Closed Won.

1. From the Elements in the Toolbox , drag an **Assignment** element onto the canvas.
2. In Label, name it Update Opportunity Fields .
3. Set the stage to Prospecting.
 - a. For Variable, select **RECORD VARIABLES** | **opportunity** | **StageName**.
 - b. For Operator, select **Equals**.
 - c. For Value, select **PICKLIST VALUES** | **Prospecting**.
4. Set Close Date to 90 days from today.
 - a. Click **Add Assignment**.
 - b. For Variable, select **RECORD VARIABLES** | **opportunity** | **CloseDate**.
 - c. For Operator, select **Equals**.
 - d. For Value, select **New Resource** and set these values.

Field	Value
Resource Type	Formula
API Name	ninetyDays
Data Type	Date

5. In Formula, leave Insert a resource... blank and enter `TODAY() + 90` in the text box.
6. Click **Done**.
7. In Value in the second row, select **FORMULAS** | **ninetyDays**.
8. Click **Done**.
9. To create the opportunity, drag a **Create Records** element onto the canvas and set these values.

Field	Value
-------	-------

Field	Value
Label	Clone Opportunity
Record Variable	RECORD VARIABLES > opportunity

10. Click **Done**.
11. Click the **node** at the bottom of Start and drag it to Update Opportunity Fields.
12. Click the **node** at the bottom of Update Opportunity Fields and drag it to Clone Opportunity.

When the opportunity is created, the ID field in that variable (`{!opportunity.Id}`) is populated with the new record's ID. We reference that value later to associate the new products with the renewal opportunity.

Clone the Opportunity Products

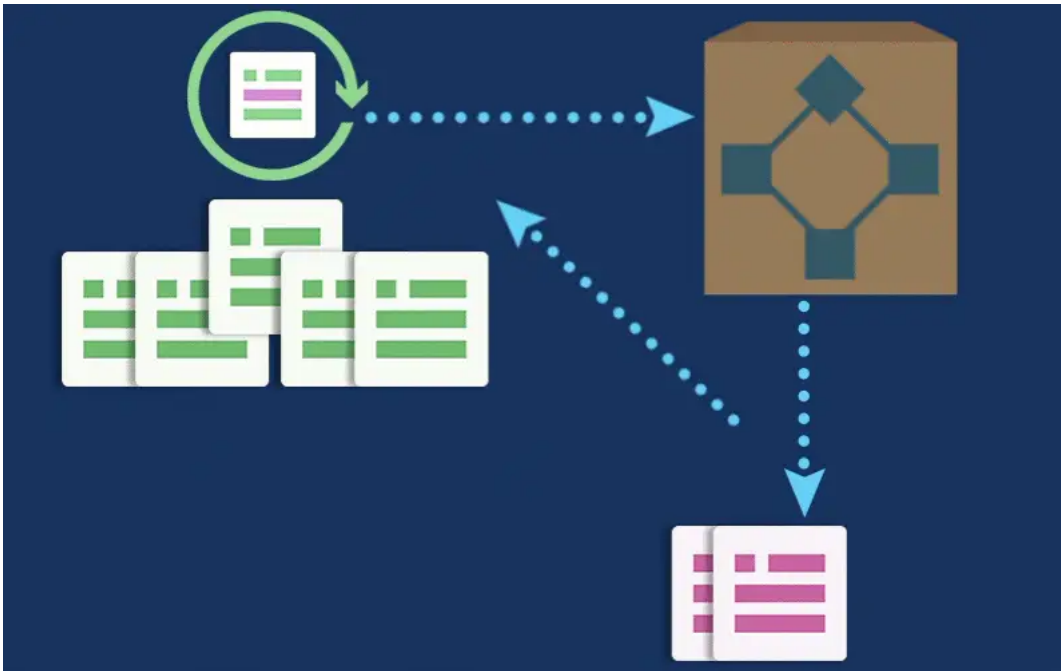
Now comes the fun part: cloning the opportunity products.

When the flow is called by the process, the `{!oppProducts_Original}` variable contains fields from the original opportunity products. Before we clone those products, we need to associate them with the renewal opportunity instead of the original opportunity and set the total price for each product to null. (Opportunity products can't have both a unit price and a total price.)

The only way to update items in a collection is to iterate over the collection with a loop. A loop tells the flow to process each item in the collection one at a time, executing the same logic on each item until the entire collection has been processed.

Each opportunity product in our `{!oppProducts_Original}` record collection should be associated with the renewal opportunity, and the total price set to null.

Each time the loop iterates, the *loop variable* represents an item in the collection. When a loop starts, the first item in the collection variable is copied into the loop variable. After the iteration is over, the loop variable is overwritten with the next item's values. And so on, until there are no items left in the collection.



Tip

- To update an item's field values inside a loop, update the loop variable. Then, before the iteration for that item finishes, add the loop variable as an item in another collection variable. If you don't do that, the changes are overwritten when the next item is loaded into the loop variable.
- Avoid adding actions, like creating or updating records, inside a loop. That's a surefire way to hit limits.

In the loop, we update each item's Opportunity ID and Total Price, and then we add the item to a new collection variable. After the loop, we use the new collection variable to create the opportunity products.

1. Drag a Loop element onto the canvas, and set these values.
 - a. In Label, name it **Iterate Over Products**.
 - b. For Collection Variable, select **RECORD COLLECTION VARIABLES | oppProducts_Original**.
This option tells the loop which collection to iterate over when assigning items to the loop variable.
 - c. For Loop Variable, select **New Resource** and set these values.

Field	Value
Resource Type	Variable
API Name	oppProduct_loop
Data Type	Record
Object	Opportunity Product

2. Click **Done**.
3. Set the Loop Variable to **RECORD VARIABLES | oppProduct_loop**.
4. Click **Done**.
5. Connect Clone Opportunity to the loop.
6. Set the loop variable's opportunity ID to the ID of the new opportunity and its total price to null.
 - a. Drag an **Assignment** element onto the canvas.
 - b. In Label, name it Update Opportunity Product .
 - c. In Set Variable Values, configure these assignments. Click **Add Assignment** to add a second line.

Variable	Operator	Value
RECORD VARIABLES > oppProduct_loop > OpportunityId	Equals	RECORD VARIABLES > opportunity > Id
RECORD VARIABLES > oppProduct_loop > TotalPrice	Equals	Leave blank

Edit Assignment

* Label

Update Opportunity Product

* API Name

Update_Oppportunity_Product

Description

Set Variable Values

Each variable is modified by the operator and value combination.

Variable

{!oppProduct_loop.OpportunityId}

Operator

Equals

Value

{!opportunity.Id}

Variable

{!oppProduct_loop.TotalPrice}

Operator

Equals

Value

Enter value or search resources...

+ Add Assignment

Cancel

- d. Click **Done**.
7. Connect the loop to Update Opportunity Product. The Select loop connector window appears.
8. Verify that **For each item in the collection** is selected in the Loop Connector field and click **Done**.
9. Add the updated loop variable to a new record collection variable.
 - a. Drag another **Assignment** element onto the canvas.
 - b. In Label, name it Add to New Collection .
 - c. For Variable, select **New Resource** and create a new resource variable with these values.

Field	Value
Resource Type	Variable

Field	Value
API Name	oppProducts_new
Data Type	Record
Object	Opportunity Product
Allow multiple values (collection)	Selected

d. Click **Done**.

e. For Variable, select **RECORD COLLECTION VARIABLES** | **oppProducts_new**.

f. For Operator, select **Add**.

g. For Value, select **RECORD VARIABLES** | **oppProduct_loop**.

h. Click **Done**.

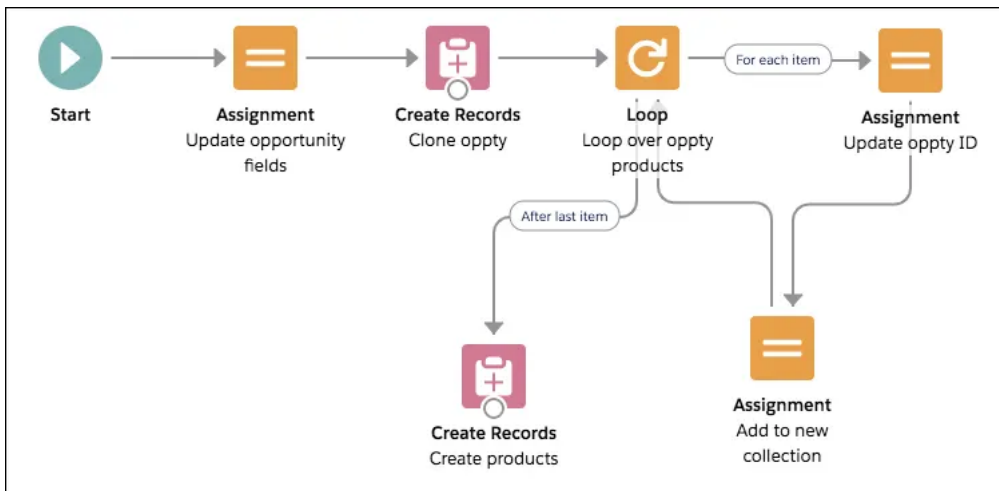
10. Connect Update Opportunity Product to Add to New Collection. Then connect Add to New Collection to the loop.

11. To clone the opportunity products, drag a **Create Records** element onto the canvas and set these values.

Field	Value
Label	Clone Products
How Many Records to Create	Select Multiple
Record Collection Variable	RECORD COLLECTION VARIABLES > oppProducts_new

12. Click **Done**.

13. Connect the loop to Clone Products.



Finish the Flow

You're nearly done! To finish the flow, save and activate it.

1. Click **Save**.
2. Save the flow, and name it **Renew Opportunity**. Be sure **Type** is set to Autolaunched Flow.
3. Click **Save**.

Close Flow Builder and activate the flow so that you can reference it in Process Builder.

Finish the Process

Now that you've finished building the flow, make sure that the process starts the flow when an opportunity is Closed Won. Remember those variables that we created at the beginning of the flow? In the process, we use them to pass values from the opportunity record and its children into the flow.

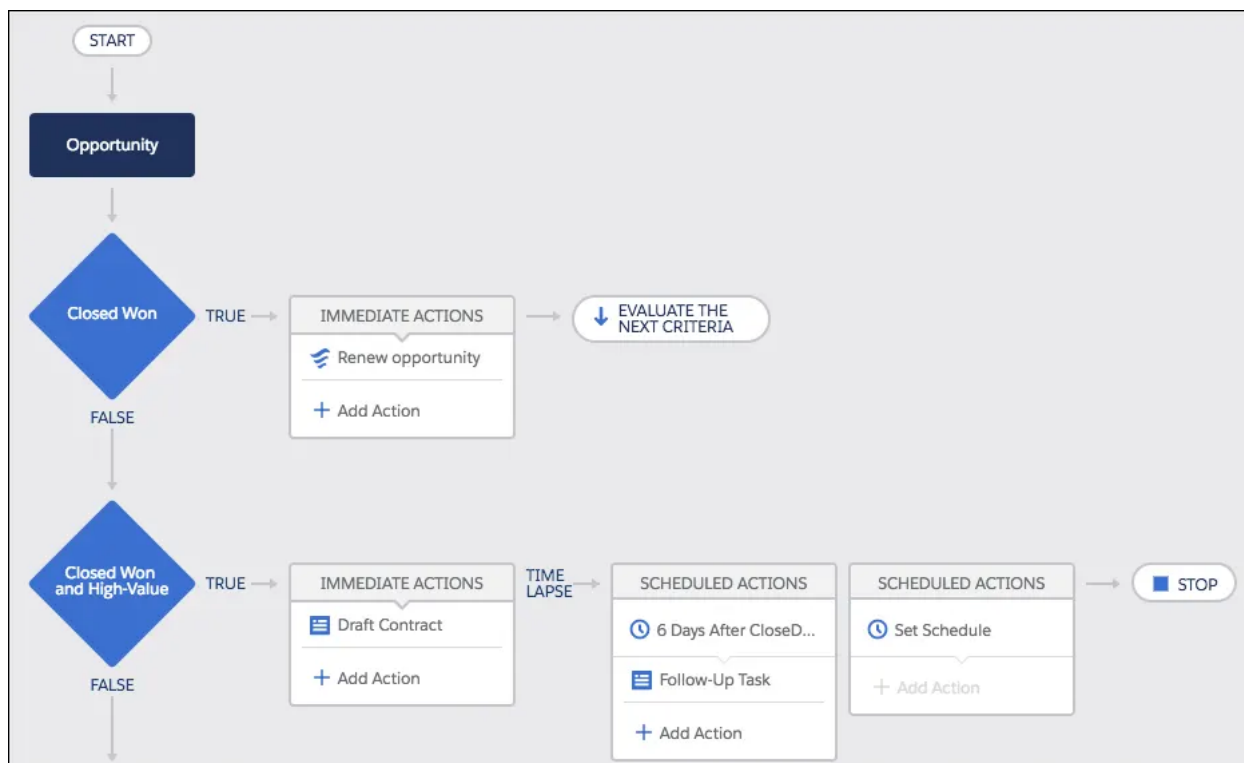
1. Open the **Closed Won Opportunities** process.

2. In the Closed Won criteria group, add an immediate action where the type is **Flows** and the name is **Create Renewal**.
3. For Flow, select **Renew Opportunity**.
4. Under Set Flow Variables, add two rows, set these values, and then click **Save**.

Variable	Type	Value
opportunity	Field Reference	Select the Opportunity record that started the process
oppProducts_original	Field Reference	Opportunity > OpportunityLineItems

When an opportunity is Closed Won, the renewal should always be created. But you need to make sure that draft contracts and follow-up tasks are created if the opportunity is also considered to be high value.

5. Drag the **Closed Won** criteria node above Closed Won and High Value.
6. For the Closed Won criteria group, click **STOP**.
7. Select **Evaluate the next criteria** and then click **Save**.



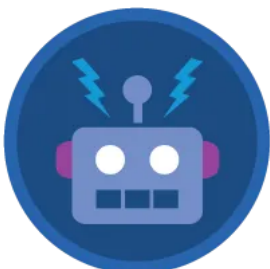
8. To start using the process, activate it.

Resources

- Trailhead Project: [Build an Account Reassignment Wizard](#)
- Trailhead Project: [Build a Workshop Management System](#)

Quiz Complete!

+25 points



Lightning Flow
100%
Progress: 100%
[View more modules](#)