

Choose Automation Tool:

Lightning Flow provides declarative process automation for every Salesforce app, experience, and portal.

Lightning Flow **includes** Process Builder and Flow Builder, to build processes and flows respectively.

LF is the product, PB and FB are the tools.

Which tool to use, when?

Process Type, Description, Available Tool(s)

- Guided visual Exp., processes that need input from user-emp or cust, **Flow Builder**.
- Behind-scenes automation, processes that get all data from your org or connected system - no user input, **Process Builder, Flow Builder, Apex** (cust code)
- Approval automation, process that determine how a record gets approved by stakeholders, **Approvals**.

Best practice to start with declarative no-code tools and work up to code solutions.

Use **Process Builder** when starting a behind scenes process automatically:

- Record is created/updated
- A platform event occurs

Use **Flow Builder** to:

- Automate a guided visual experience.
- Add functionality for behind-scenes process on top of PB. Call the flow from the process
- Start behind-scenes process when a user clicks something.

Use **Apex** when you need more functionality than the previous two offer. Build it as invocable Apex methods. Then call as an Apex action in the process or Apex action element in the flow.

PB and FB offer more features and can do more than **Workflow**. They also include new functionality for behind the scenes automation.

**Process Builder** in more detail:

Point/Click tool that lets you automate **if/then** processes and see a graphical representation as you build. (LucidChart, for example)

A **Trigger** identifies when to run the process(only one per process). **Criteria nodes** (unlimited#) on that process controls whether or not the process executes the associated actions. If a criteria is not met, the process moves to the next Criteria Node in the process.

Each Criteria node can:

- Set filter conditions
- Enter a custom formula (validation rules, for example)
- Opt out of criteria and always execute the associated actions.

## Process Actions

When a criteria node evals true, process executes actions (**immediate**) or waits to execute at a scheduled time (**scheduled**). A scheduled action will check the criteria node still evals to true before executing. Schedule based on specific date/time field OR time that the process ran (ex: 3 days from now...)

Some examples of process actions:

- Create record. Update record that triggered process OR any related record(s)
- Submit record for approval
- Send emails using a template, Post to a feed

Process Types and Triggers: Type - Trigger

- Record Change - Triggered when record is created or edited
- Invocable (method) - Triggered when called by another process
- Platform Event - Triggered when platform event message is received (Listener?)

**Flow Builder** in more detail:

\*remember, **Lightning Flow** - is the product for building, managing, and running flows AND processes. **Flow Builder** is the tool for building flows. A **Flow** is an application that automates a bus process by collecting data and performing an action in your SF org or ext sys.

tl;dr: LF( Flow => {flow}). :)

Flow Builder has replaced legacy 'Cloud Flow Designer' feature. 'Visual Workflow' is a retired feature and is in comparison/replaced by Lightning Flow. VW(CFD => {flow})

Three Building Blocks of a flow:

1. Elements - These are pieces of the flow you click/drag to the canvas from the toolbox.
2. Connectors - these are lines that define the path the flow will take when running. Outline the sequence of element execution/actions.
3. Resources - containers (aka fields/entries) that represent field values or formulas. What are you referencing within your org during the flow? What will end up getting modified, how, etc. Can also be thought of as the details to an element/action.

## Elements:

Four categories of elements.

1. Screen - These are elements for displaying data to or obtaining data from users. Think **forms**.
2. Logic- These control the flow. Think branching, looping, update, or wait for x.

3. Actions - Perform something in Salesforce when you have necessary info, such as user input. Think CRUDs to fields/objects, make chatter posts, submit something for approvals, send emails etc. Need something more, make a call to Apex code.
4. Integrations - Connect flow to an ext database. This is done with core actions or Apex actions.
  - a. Core Actions let you make requests w/out going through Salesforce server.
  - b. FB offers some tie-ins to platform events. Publish event with a Create Records element. Subscribe to platform events with a Pause element.

Keyboard shortcuts when in Flow Builder:

CTRL = - zoom in

CTRL - zoom out

CTRL 0 - zoom to fit

CTRL 1 - zoom to view

Hold SHFT and click multiple elements to select more than one.

Making a Flow: Setup, Flows, New Flow - choose type, Create.

Once a flow is finished, There are two(2) runtime experiences that determine look/feel when someone runs a flow. To have the flow blend in with Lightning Experience make sure Lightning Runtime is enabled in your org. Here's how....

- Setup, 'Automation', select 'Process Automation Settings'
- Verify 'Enable Lightning runtime for flows' is selected. SAVE.

To Distribute the Flow, we need to get it to correct users by adding it to the Home page. There are other ways to distribute as well.

Example steps: click flow, click Activate button in top bar. Create a Home page by Setup, builder search, select Lightning App Builder. Click 'New' from Lightning pages list. Select Home Page, Next. Click Clone Salesforce Default Page from tab option, We then drag the flow onto the home page (after selecting layout type) and save/activate it. Set the Home Page as default so we can see it in action. Click Setup Menu icon, view all, select Home

A **flow interview** is a running **instance** of a flow. When you distribute a flow, users interact with an interview(instance) of that flow.

### Using Process Builder AND Flow Builder together:

Since PB can't do everything (such as post to a comm feed, delete records, submit for approval, complex logic) you will need to do things in FB and add a flow action to the process. The next step is writing Apex code to do something and add an Apex action to the process.

Ex: auto create renewal opportunities when a current opp is won and closed(Closed Won). Should clone the original opp. PB can clone records, but we need to clone products and

associate with the renewal opp. PB can't grab ID of the created record and use it elsewhere but a flow can. So, make a flow that clones the opp and products, and call the flow within the process whenever an opp is closed.

**\*Autolaunched flows** run in the bg like a process. They can't have screens, so no user interaction. You can call them from process and Apex classes.

To accomplish the above example, we create a criteria in the process where it checks to see if the opp has been set to Closed Won. Then we switch to Flows. Create a new autolaunched flow. This Flow will need data from the process - orig opp and its opp products.

**Flow Variables** (one of many **\*Resource types\*** see above) come in four types:

1. Variable - a single value (string/bool/num etc)
2. Collection Variable - Multiple values of the same data type (ex: array of strings or nums)
3. Record variable - A set of field values for a single record. (ex: rating, Id, and Name from an account object)
4. Record Collection Variable - A set of field values for multiple records of the same object type (ex: rating, ID, and Name of all or multiple account objects)

We'll create record var and rec collection var and we'll pass the data into these during the process so they must allow input!

<Toolbox, Manager, New Resource, type=variable> to get started. (this is an in-depth process, reference the learning materials for all necessary steps: lightning flow module, combine power of process builder and flow builder)

## **APPROVALS:**

Used to define approval processes, automate actions/changes to fields on approval/rejection  
They outline the steps needed for an approval and actions taken based on approval/rejection