Flow General Notes

Generalities

As of writing this page, August 10th 2023, Flows are primary source of automation on the Salesforce platform. We left this sentence because the earlier iteration (from 2021) identified that Flows would replace Process Builder and we like being right.

> It is very important to note that Flows have almost nothing to do, complexity-wise, with Workflows, Process Builder, or Approval Processes. Where the old tools did a lot of (over)-simplifying for you, Flow exposes a lot of things that you quite simply never had to think about before, such as execution context, DML optimization, batching, variables, variable passing, etc.
>
> So if you are an old-timer upgrading your skills, note that **a basic understanding of programming (batch scripting is more than enough) helps a lot with Flow**.
> If you're a newcomer to Salesforce and you're looking to learn Flow, same comment - this is harder than most of the platform (apart from Permissions) to learn and manipulate. This is normal.

Intended Audience

These conventions are written for all types of Salesforce professionals to read, but the target audience is the administrator of an organization. If you are an ISV, you will have considerations regarding packaging that we do not, and if you are a consultant, you should ideally use whatever the client wants (or the most stringent convention available to you, to guarantee quality).

On Conventions

As long as we're doing notes: conventions are opinionated, and these are no different. Much like you have different APEX trigger frameworks, you'll find different conventions for Flow. These specific conventions are made to be maintainable at scale, with an ease of modification and upgrade. This means that they by nature include boilerplate that you might find redundant, and specify very strongly elements (to optimize cases where you have hundreds of Flows in an organization). **This does not mean you need to follow everything.** A reader should try to understand *why* the conventions are a specific way, and then decide whether or not this applies to their org.

> At the end of the day, as long as you use **any** convention in your organization, we're good. This one, another one, a partial one, doesn't matter. Just structure your flows and elements.

On our Notation

Finally, regarding the naming of sub-elements in the Flows: we've had conversations in the past about the pseudo-hungarian notation that we recommend using. To clarify: we don't want to use Hungarian notation. We do so because Flow still doesn't split naming schemes between variables, screen elements, or data manipulation elements. This basically forces you to use Hungarian notation so you can have a `var_boolUserAccept` and a `S01_choiceUserAccept` (a variable holding the result of whether a user accepts some conditions, and the presentation in radio buttons of said acceptance), because you can't have two elements just named `UserAccept` even if technically they're different.

On custom code, plugins, and unofficialSF

On another note: Flow allows you to use custom code to extend its functionality. We define "custom code" by any LWC, APEX Class, and associated that are written by a human and plug into flow. We recommend using as little of these elements as possible, and as many as needed. **This includes UnofficialSF**.

> Whether you code stuff yourself, or someone else does it for you, Custom Code always requires audit and maintenance. Deploying UnofficialSF code to your org basically means that you own the maintenance and audit of it, much like if you had developed it yourself. We emit the same reservations as using any piece of code on GitHub - if you don't know what it does **exactly**, you shouldn't be using it. This is because any third-party code is **not part of your MSA with Salesforce, and if it breaks, is a vector of attack, or otherwise negatively impacts your business, you have no official support or recourse.**
>
> This is not to say that these things are not great, or value-adding - but you are (probably) an admin of a company CRM, which means your first consideration should be **user data and compliance**, and ease of use coming second.

Bonus useless knowledge: Flows themselves are just an old technology that Salesforce released in 2010: Visual Process Manager. That itself is actually just a scripting language: "The technology powering the Visual Process Manager is based on technology acquired from Informavores, a call scripting startup Salesforce bought last year." (2009) Source

---

Revision #10
Created 2021-01-13 13:41:44 UTC by Windyo
Updated 2023-08-16 08:19:51 UTC by Windyo