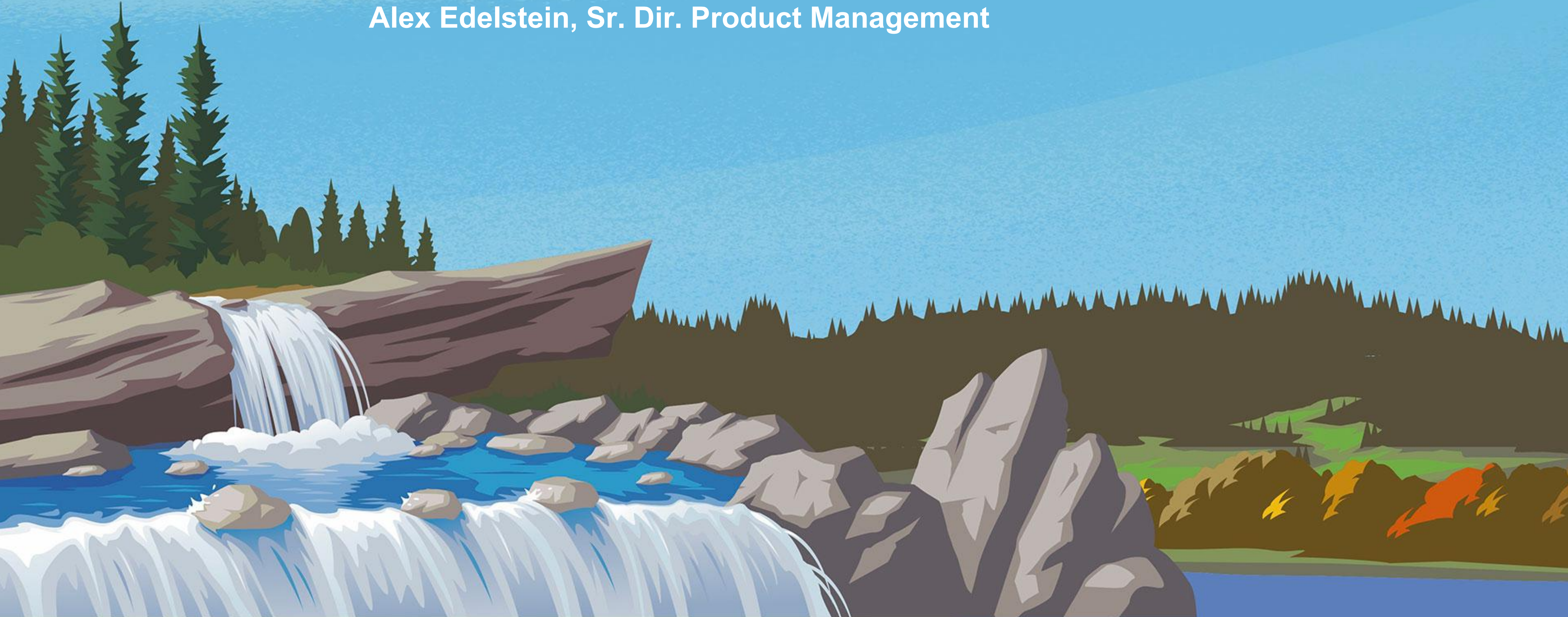
The Flow Generator Pattern

Alex Edelstein, Sr. Dir. Product Management



What’s a Flow Generator?

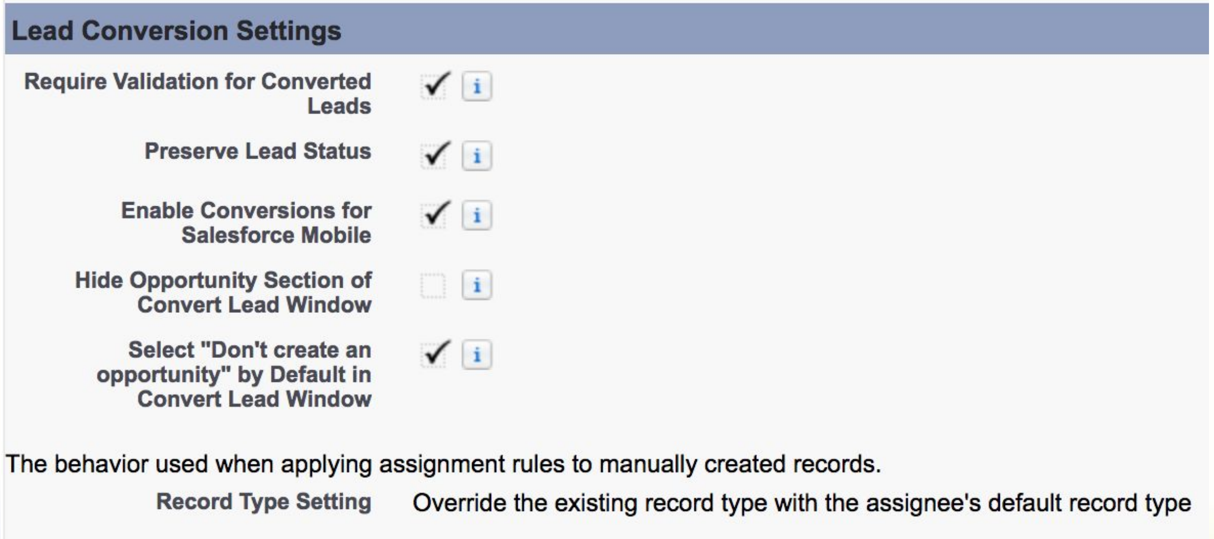
A factory that makes it easier to generate custom versions of a flow.

Use Case - Lead Conversion

Salesforce users want more ability to configure this critical process.

Convert Leads in Classic

Some customization available:

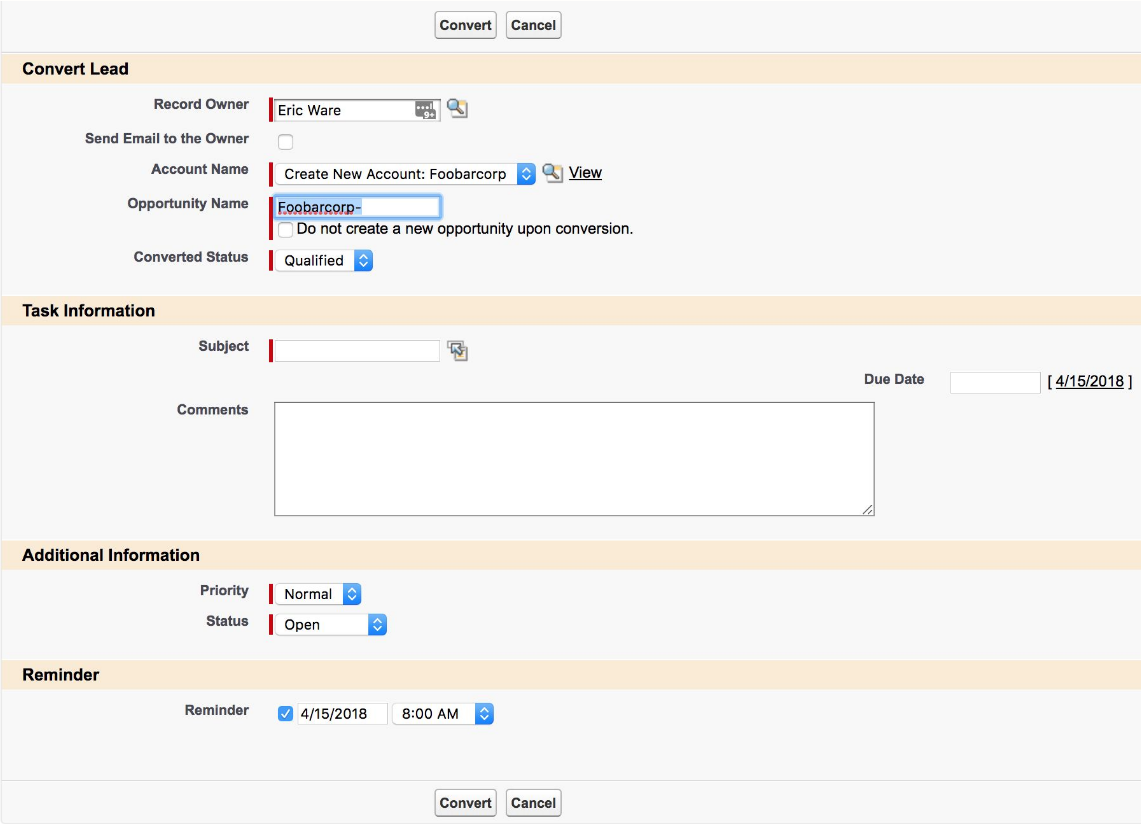


Lead Conversion Settings

Require Validation for Converted Leads	✓	i
Preserve Lead Status	✓	i
Enable Conversions for Salesforce Mobile	✓	i
Hide Opportunity Section of Convert Lead Window	□	i
Select "Don't create an opportunity" by Default in Convert Lead Window	✓	i

The behavior used when applying assignment rules to manually created records.

Record Type Setting Override the existing record type with the assignee's default record type



Convert Lead

Record Owner	Eric Ware
Send Email to the Owner	□
Account Name	Create New Account: Foobarcorp View
Opportunity Name	Foobarcorp- □ Do not create a new opportunity upon conversion.
Converted Status	Qualified

Task Information

Subject	
Due Date	[4/15/2018]
Comments	

Additional Information

Priority	Normal
Status	Open

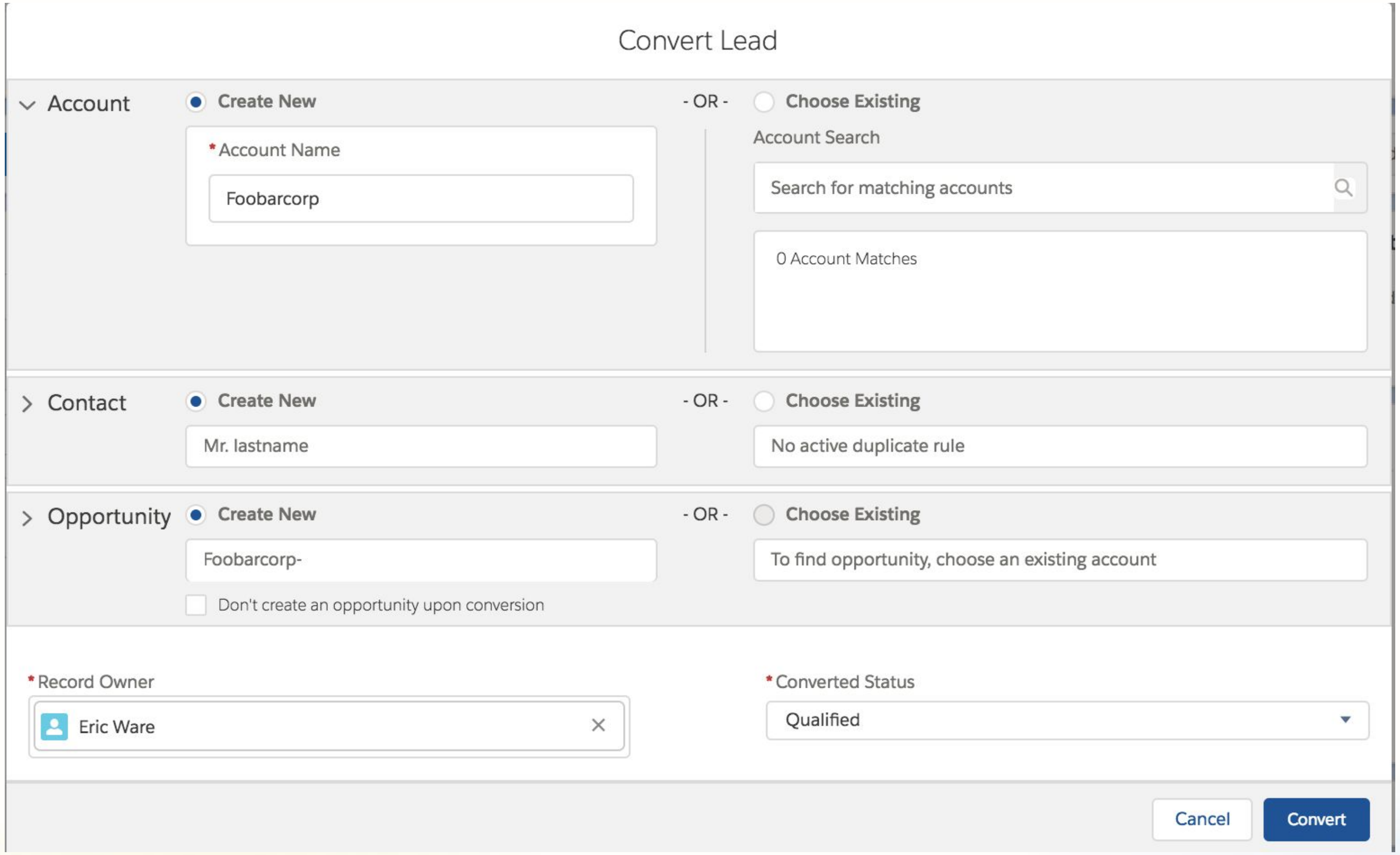
Reminder

Reminder	☑ 4/15/2018 8:00 AM

Lightning Version Has a Different Mix

With a one-page-fits-all solution, you have to make tradeoffs:

- **Can no longer add a Task**
- **Can no longer send email to owner**



Convert Lead

Account ● Create New - OR - ○ Choose Existing

*Account Name: Foobarcorp

Account Search: Search for matching accounts

0 Account Matches

Contact ● Create New - OR - ○ Choose Existing

Mr. lastname

No active duplicate rule

Opportunity ● Create New - OR - ○ Choose Existing

Foobarcorp-

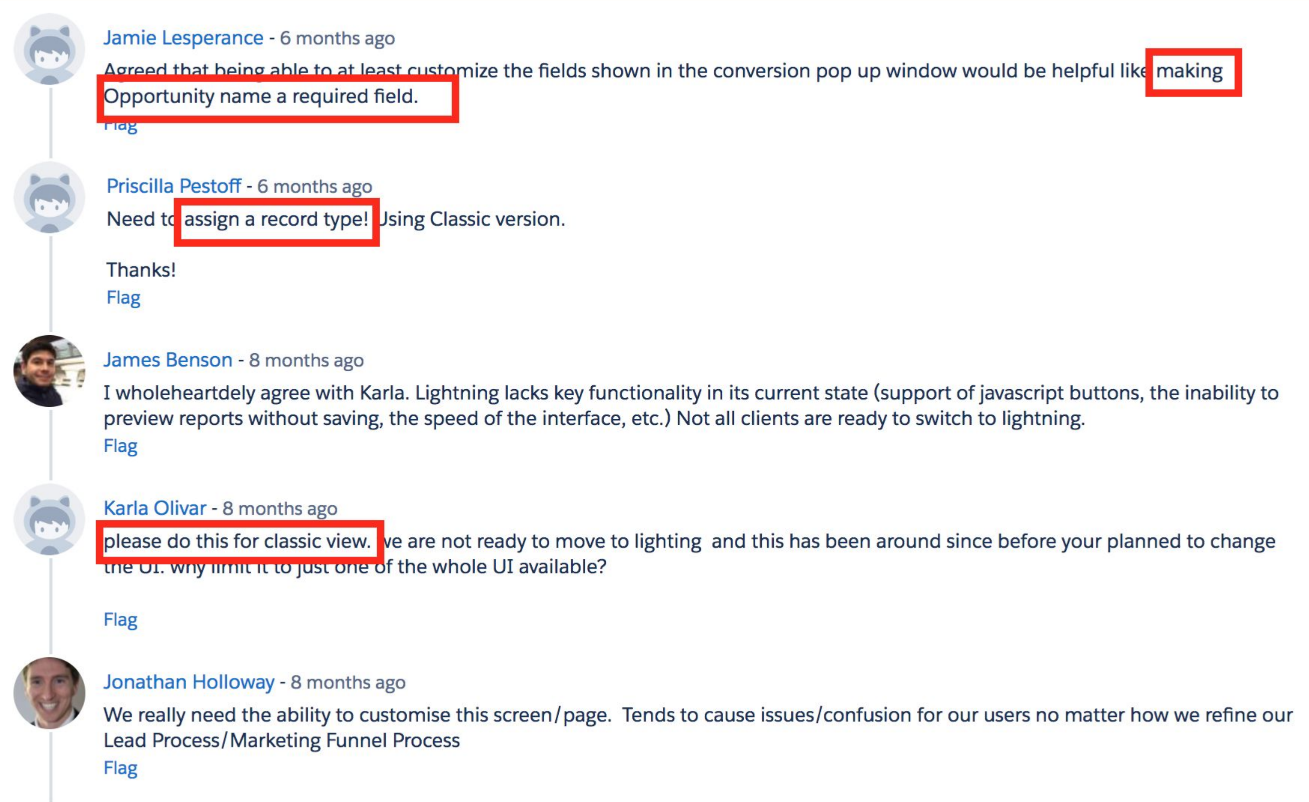
To find opportunity, choose an existing account

□ Don't create an opportunity upon conversion

*Record Owner: Eric Ware

*Converted Status: Qualified

People Want More Customization



Jamie Lesperance - 6 months ago

Agreed that being able to at least customize the fields shown in the conversion pop up window would be helpful like making Opportunity name a required field.

Flag

Priscilla Pestoff - 6 months ago

Need to assign a record type! Using Classic version.

Thanks!

Flag

James Benson - 8 months ago

I wholeheartdely agree with Karla. Lightning lacks key functionality in its current state (support of javascript buttons, the inability to preview reports without saving, the speed of the interface, etc.) Not all clients are ready to switch to lightning.

Flag

Karla Olivar - 8 months ago

please do this for classic view. we are not ready to move to lighting and this has been around since before your planned to change the UI. why limit it to just one of the whole UI available?

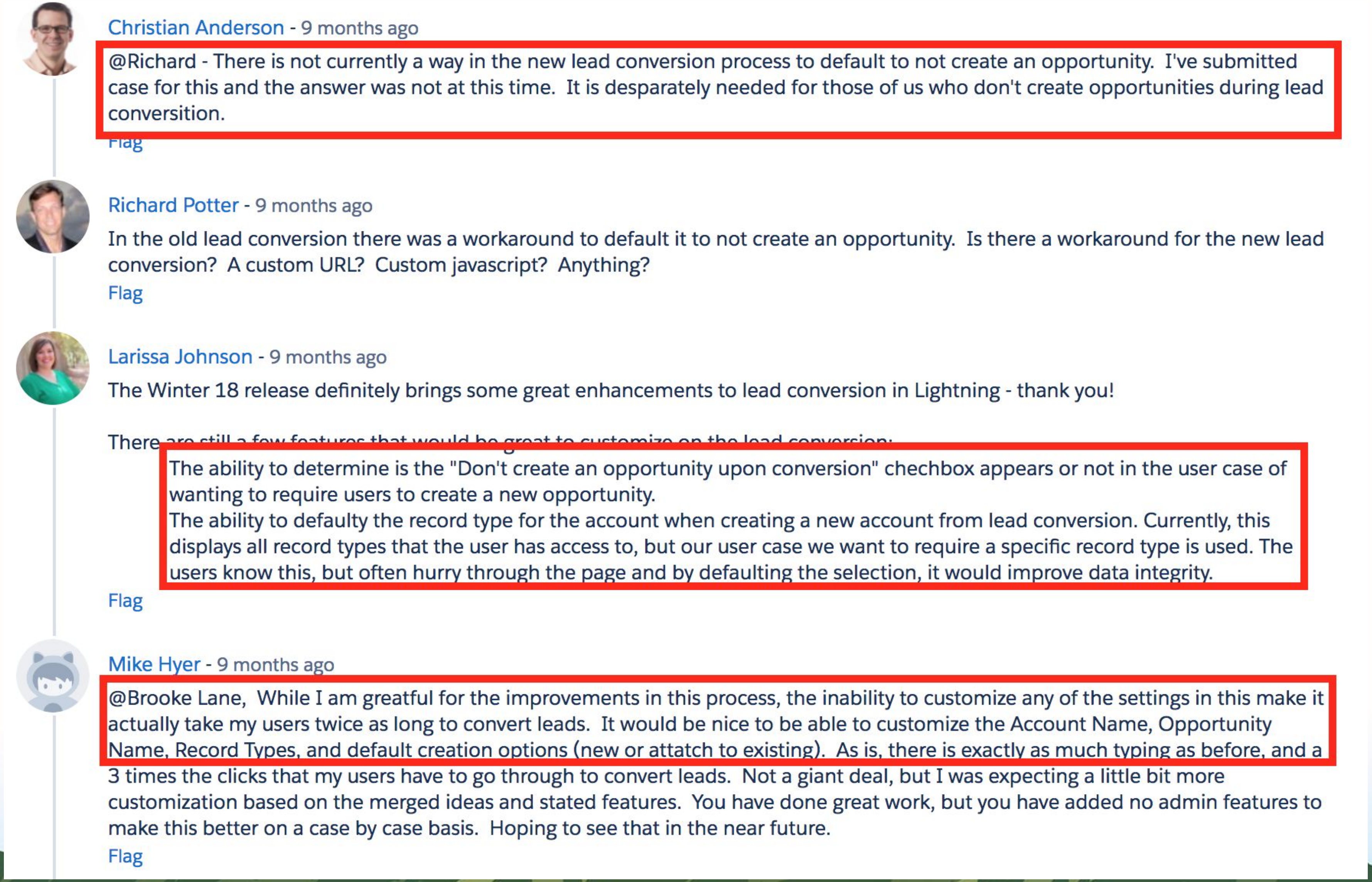
Flag

Jonathan Holloway - 8 months ago

We really need the ability to customise this screen/page. Tends to cause issues/confusion for our users no matter how we refine our Lead Process/Marketing Funnel Process

Flag

People Want More Customization



Christian Anderson - 9 months ago

@Richard - There is not currently a way in the new lead conversion process to default to not create an opportunity. I've submitted case for this and the answer was not at this time. It is desparately needed for those of us who don't create opportunities during lead conversition.

Flag

Richard Potter - 9 months ago

In the old lead conversion there was a workaround to default it to not create an opportunity. Is there a workaround for the new lead conversion? A custom URL? Custom javascript? Anything?

Flag

Larissa Johnson - 9 months ago

The Winter 18 release definitely brings some great enhancements to lead conversion in Lightning - thank you!

There are still a few features that would be great to customize on the lead conversion:

The ability to determine is the "Don't create an opportunity upon conversion" chechbox appears or not in the user case of wanting to require users to create a new opportunity.

The ability to defaulty the record type for the account when creating a new account from lead conversion. Currently, this displays all record types that the user has access to, but our user case we want to require a specific record type is used. The users know this, but often hurry through the page and by defaulting the selection, it would improve data integrity.

Flag

Mike Hyer - 9 months ago

@Brooke Lane, While I am greatful for the improvements in this process, the inability to customize any of the settings in this make it actually take my users twice as long to convert leads. It would be nice to be able to customize the Account Name, Opportunity Name, Record Types, and default creation options (new or attatch to existing). As is, there is exactly as much typing as before, and a 3 times the clicks that my users have to go through to convert leads. Not a giant deal, but I was expecting a little bit more customization based on the merged ideas and stated features. You have done great work, but you have added no admin features to make this better on a case by case basis. Hoping to see that in the near future.

Flag

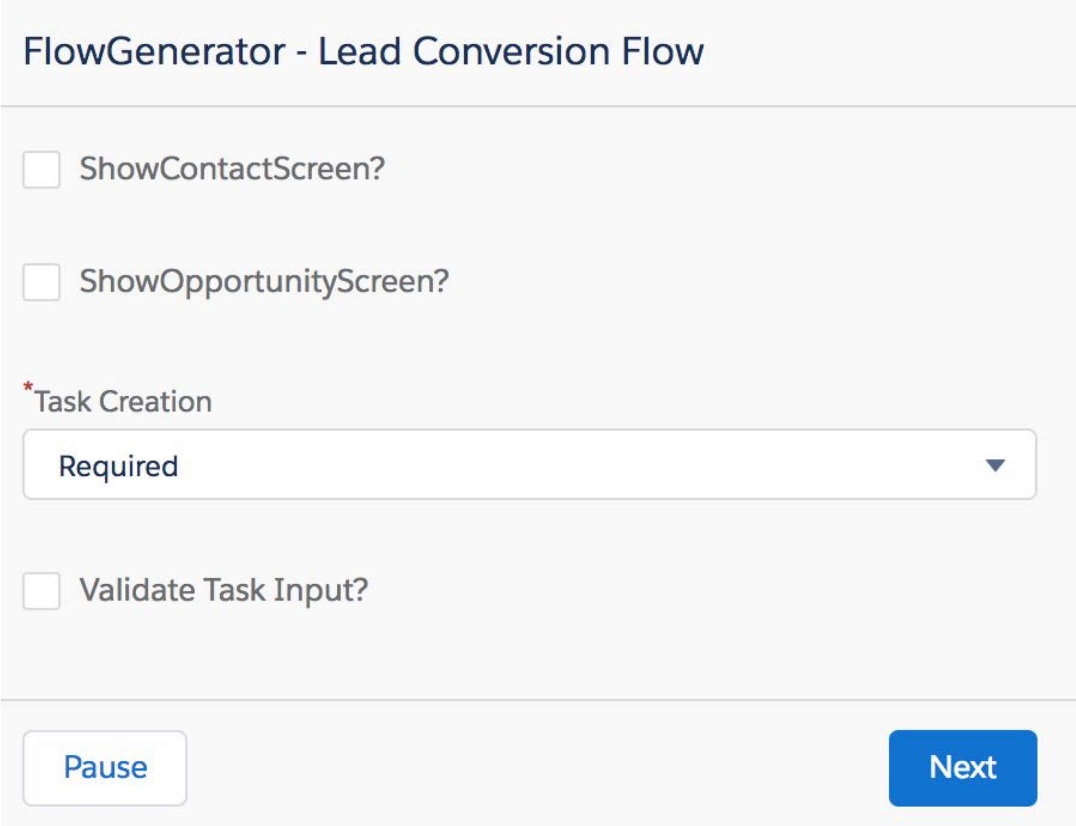
We’ve built a Flow Generator that Creates Customized Lead Conversion Flows

We’ll start by running the Flow Generator and creating two different variations of Lead Conversion

The Flow Generator is expected to be run by an Admin, while the custom Lead Conversion flows it produces are expected to be run by salespeople

Running the Flow Generator - 1

The admin specifies the customizations that they want.



FlowGenerator - Lead Conversion Flow

□ ShowContactScreen?

□ ShowOpportunityScreen?

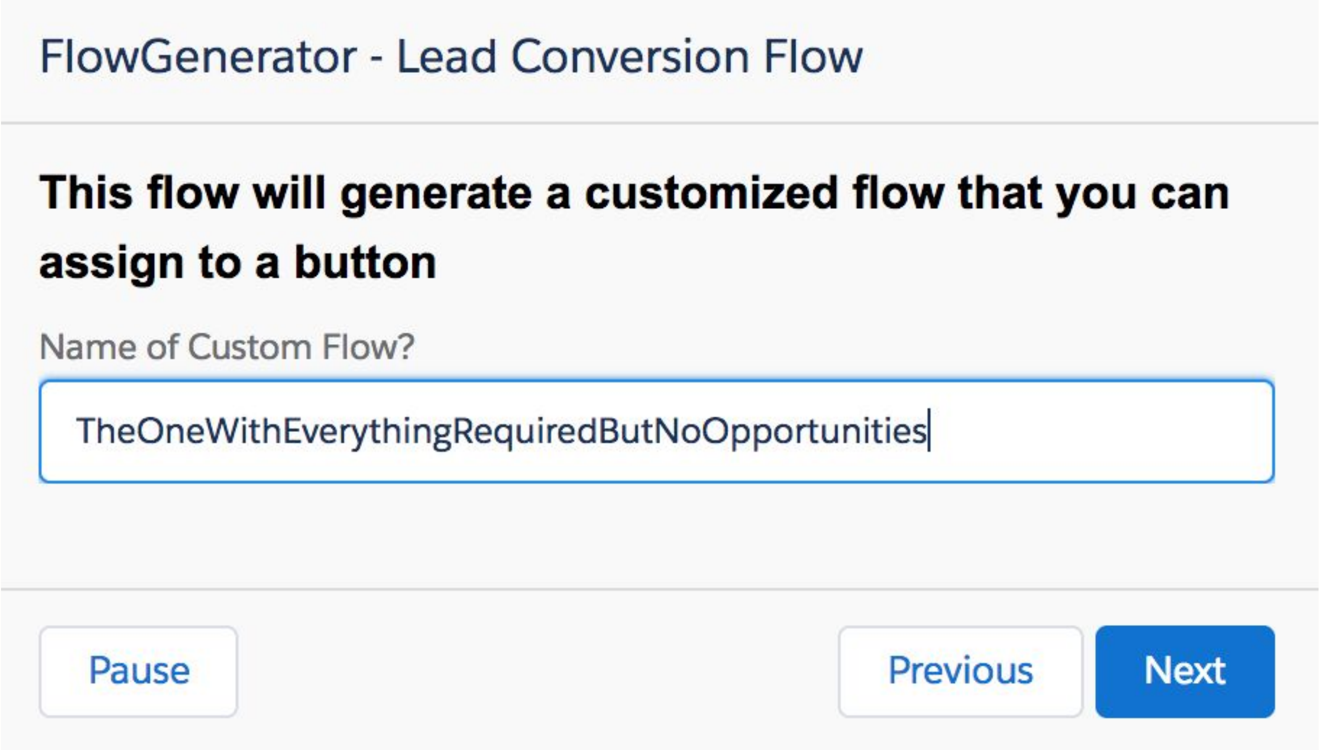
*Task Creation

Required ▾

□ Validate Task Input?

Running the Flow Generator - 2

The pick a name for the custom flow that's about to be generated.



FlowGenerator - Lead Conversion Flow

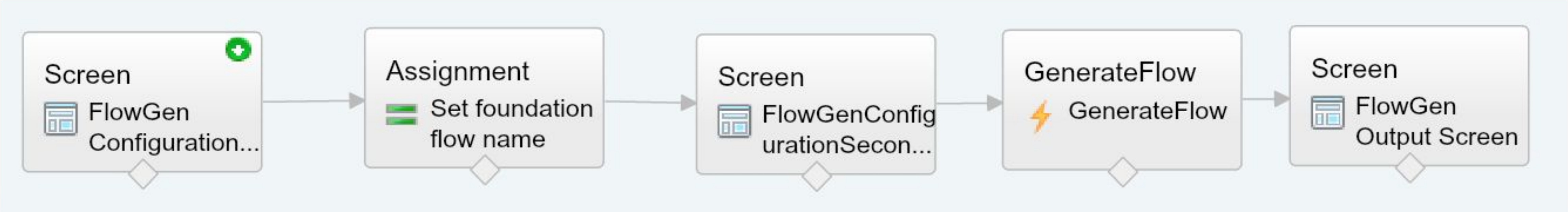
This flow will generate a customized flow that you can assign to a button

Name of Custom Flow?

TheOneWithEverythingRequiredButNoOpportunities

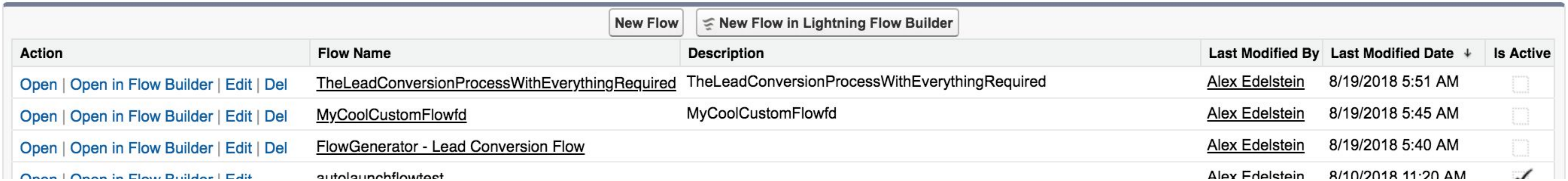
Running the Flow Generator - 3

After that, the new flow is generated. The work is done by a Flow Action called GenerateFlow that invokes an Apex Class



Running the Flow Generator - 3

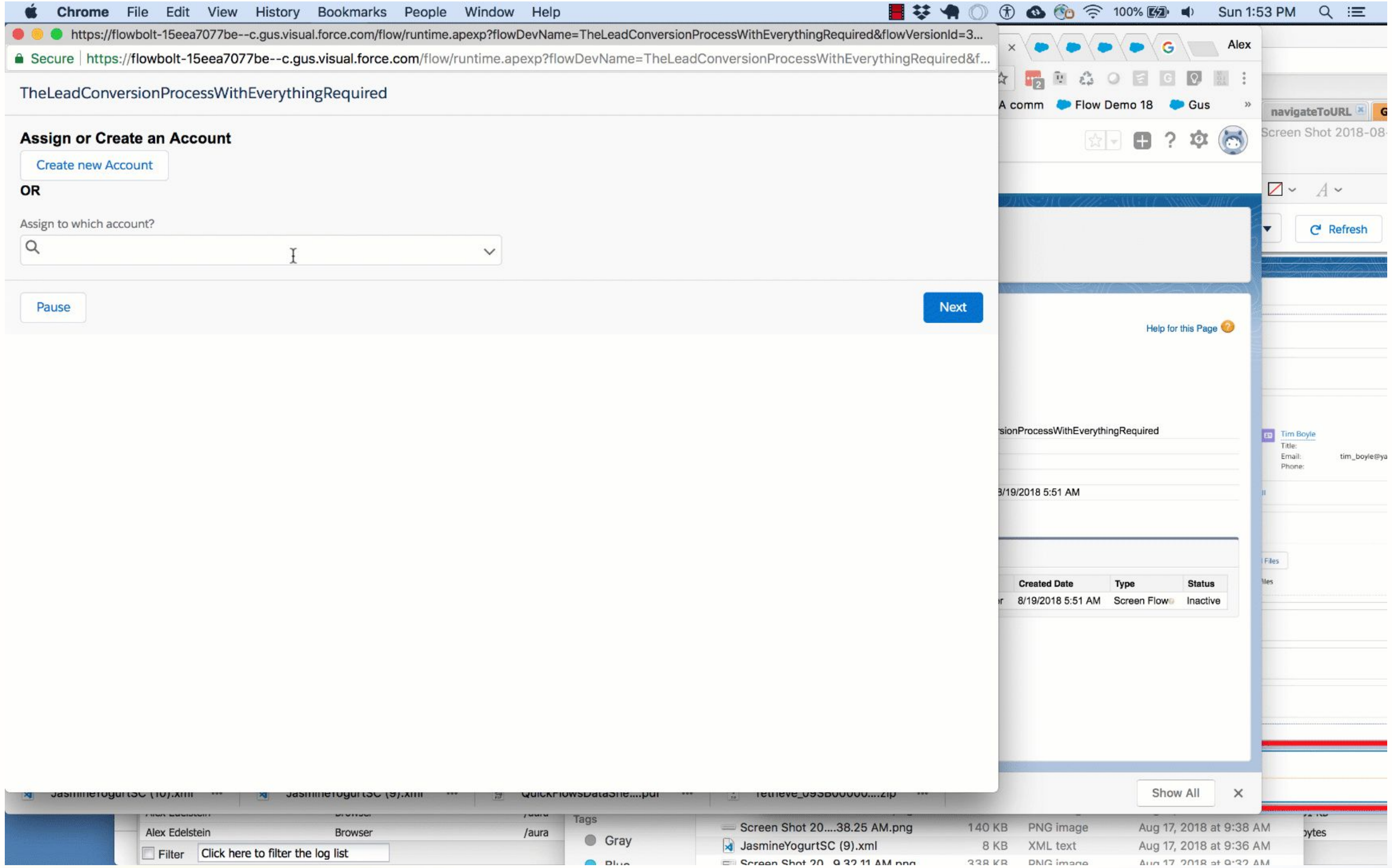
You can immediately spot the newly created Lead Conversion flow in Setup:



Action	Flow Name	Description	Last Modified By	Last Modified Date ↓	Is Active
Open | Open in Flow Builder | Edit | Del	TheLeadConversionProcessWithEverythingRequired	TheLeadConversionProcessWithEverythingRequired	Alex Edelstein	8/19/2018 5:51 AM	□
Open | Open in Flow Builder | Edit | Del	MyCoolCustomFlowfd	MyCoolCustomFlowfd	Alex Edelstein	8/19/2018 5:45 AM	□
Open | Open in Flow Builder | Edit | Del	FlowGenerator - Lead Conversion Flow		Alex Edelstein	8/19/2018 5:40 AM	□
Open | Open in Flow Builder | Edit	autolaunchflowtest		Alex Edelstein	8/10/2018 11:20 AM	✓

Running the Flow Generator - 3

Running the Flow shows the customizations



https://flowbolt-15eea7077be--c.gus.visual.force.com/flow/runtime.apexp?flowDevName=TheLeadConversionProcessWithEverythingRequired&flowVersionId=3...

TheLeadConversionProcessWithEverythingRequired

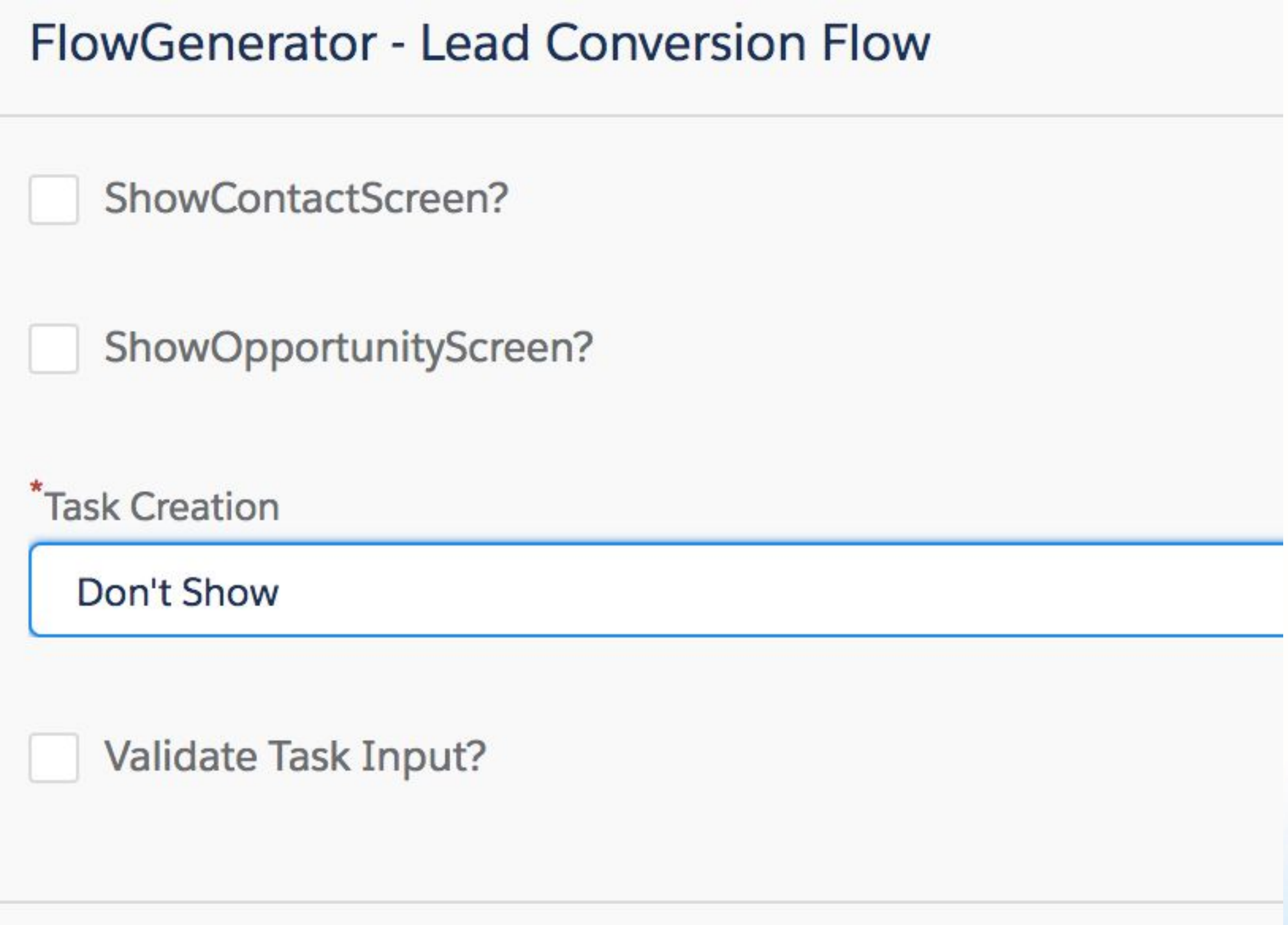
Assign or Create an Account

OR

Assign to which account?

Running the Flow Generator - 3

To show the variation, let’s run the same Flow Generator but create a minimalist Lead Conversion process



FlowGenerator - Lead Conversion Flow

□ ShowContactScreen?

□ ShowOpportunityScreen?

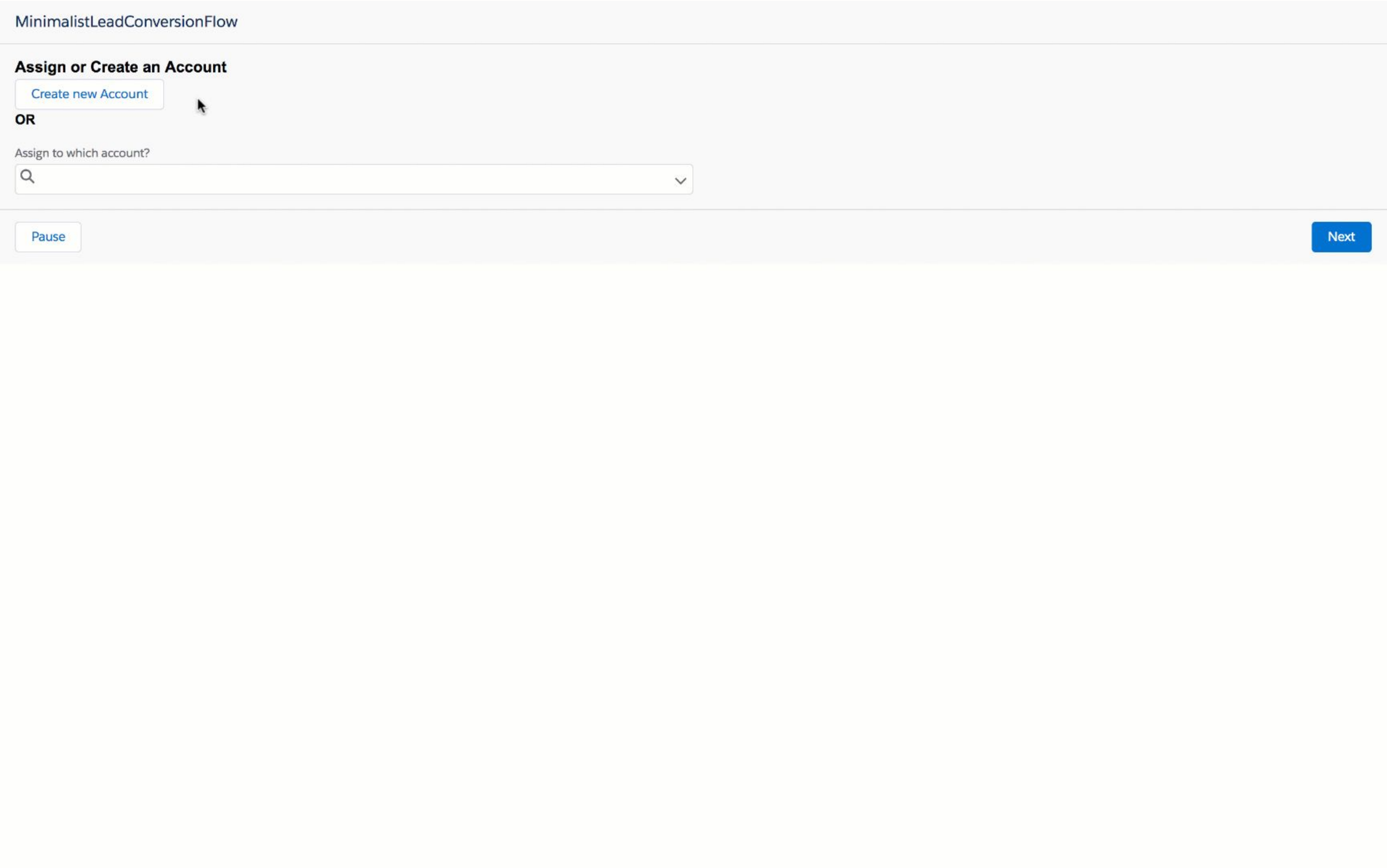
*Task Creation

Don't Show

□ Validate Task Input?

Running the Minimal Lead Conversion

Running the Flow shows the customizations



MinimalistLeadConversionFlow

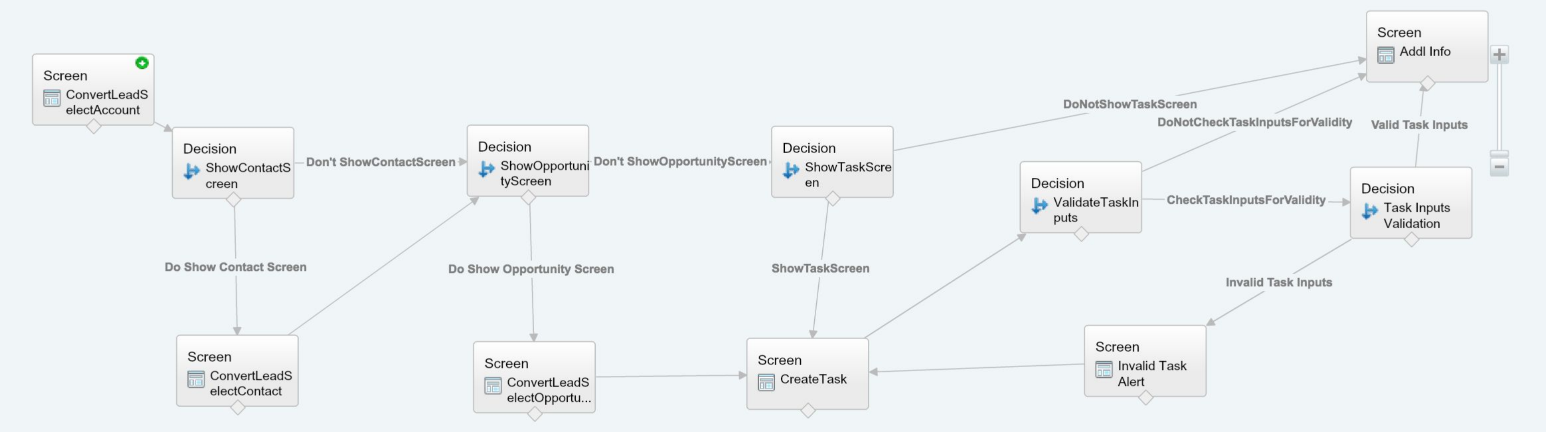
Assign or Create an Account

OR

Assign to which account?

How Does it Work?

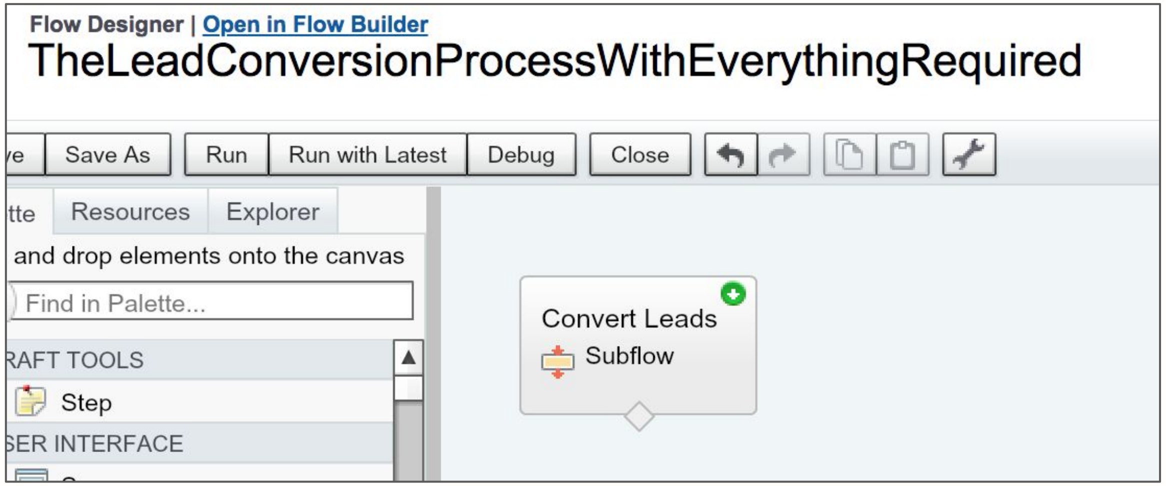
We start with a “foundation flow” that contains decision points enabling every combination of lead conversion



- When this flow runs, the branching decisions will be driven by values passed in at the beginning of the flow.
- The user running this flow will not make any decisions themselves

How Does it Work?

When the Flow Generator builds a custom flow, it creates a single subflow call that passes in the settings provided in the Flow Generator's screens by the admin



Flow Designer | Open in Flow Builder

TheLeadConversionProcessWithEverythingRequired

- This is the only element of each of the custo flows!
- All the work is done by the Flow Generator, which assembles the decisions, and the foundation flow, which has all of the logic.

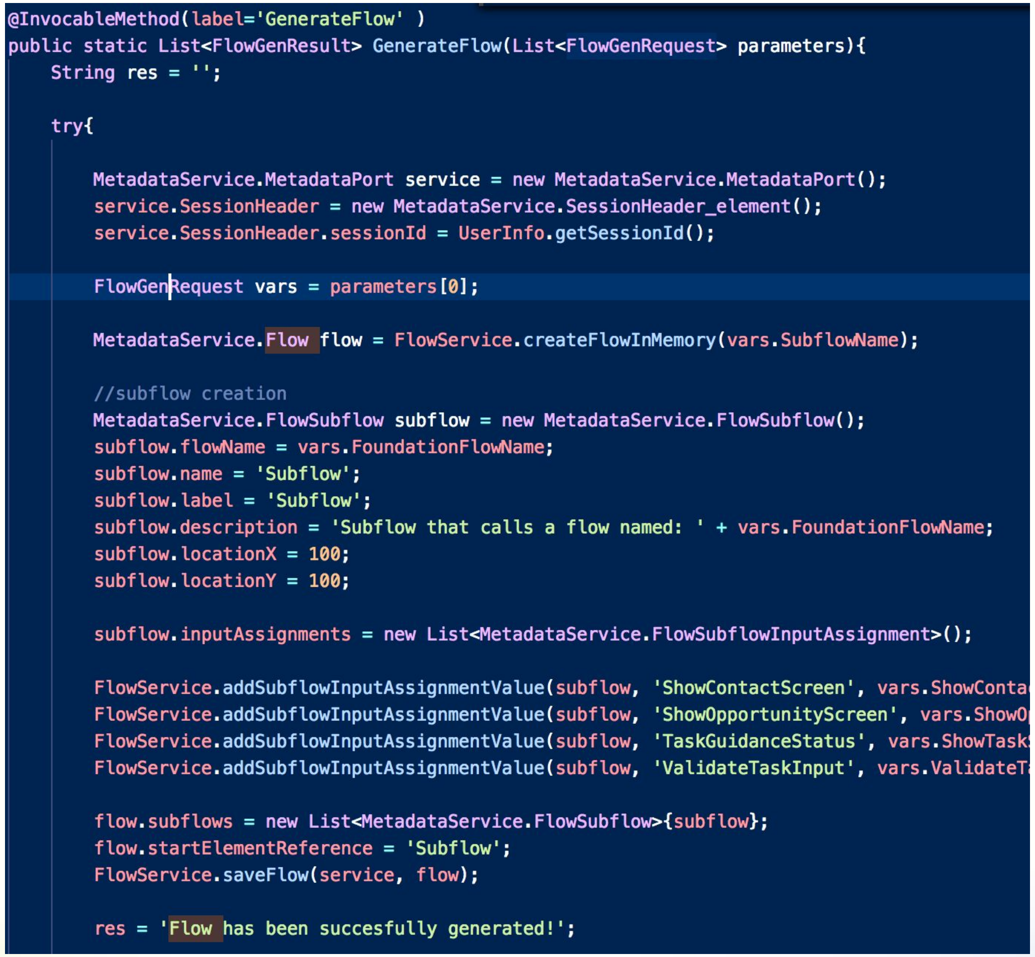
What’s in the GenerateFlow Apex Class

GenerateFlow calls an Apex Class

The Apex Class programmatically:

1) creates a new flow
2) adds a subflow element
3) adds InputAssignmentValues for each configuration created earlier in the Flow

UNOFFICIAL Code available at github.com/alexed1/flowgenerator



```
@InvocableMethod(label='GenerateFlow' )
public static List<FlowGenResult> GenerateFlow(List<FlowGenRequest> parameters){
    String res = '';

    try{

        MetadataService.MetadataPort service = new MetadataService.MetadataPort();
        service.SessionHeader = new MetadataService.SessionHeader_element();
        service.SessionHeader.sessionId = UserInfo.getSessionId();

        FlowGenRequest vars = parameters[0];

        MetadataService.Flow flow = FlowService.createFlowInMemory(vars.SubflowName);

        //subflow creation
        MetadataService.FlowSubflow subflow = new MetadataService.FlowSubflow();
        subflow.flowName = vars.FoundationFlowName;
        subflow.name = 'Subflow';
        subflow.label = 'Subflow';
        subflow.description = 'Subflow that calls a flow named: ' + vars.FoundationFlowName;
        subflow.locationX = 100;
        subflow.locationY = 100;

        subflow.inputAssignments = new List<MetadataService.FlowSubflowInputAssignment>();

        FlowService.addSubflowInputAssignmentValue(subflow, 'ShowContactScreen', vars.ShowConta
        FlowService.addSubflowInputAssignmentValue(subflow, 'ShowOpportunityScreen', vars.ShowO
        FlowService.addSubflowInputAssignmentValue(subflow, 'TaskGuidanceStatus', vars.ShowTask
        FlowService.addSubflowInputAssignmentValue(subflow, 'ValidateTaskInput', vars.ValidateT

        flow.subflows = new List<MetadataService.FlowSubflow>{subflow};
        flow.startElementReference = 'Subflow';
        FlowService.saveFlow(service, flow);

        res = 'Flow has been succesfully generated!';
```

Analysis

- Creating the custom apex class at the heart of the Flow Generator is not trivial, but integrators and developers can use this approach to deliver highly customizable solution
- Note that the Foundation Flow can be in a managed package, allowing IP to be screened from visibility, even as clients are given a way to carry out declarative field customizations
- A logical extension of this project will enable Flow Generators to be created by running a Flow that generates Flow Generators! It will take the admin config choices as key value pairs, eliminating the need to customize the apex code each time.