

Fancy Picklist/Dropdown List Choices in Flow

Use Case

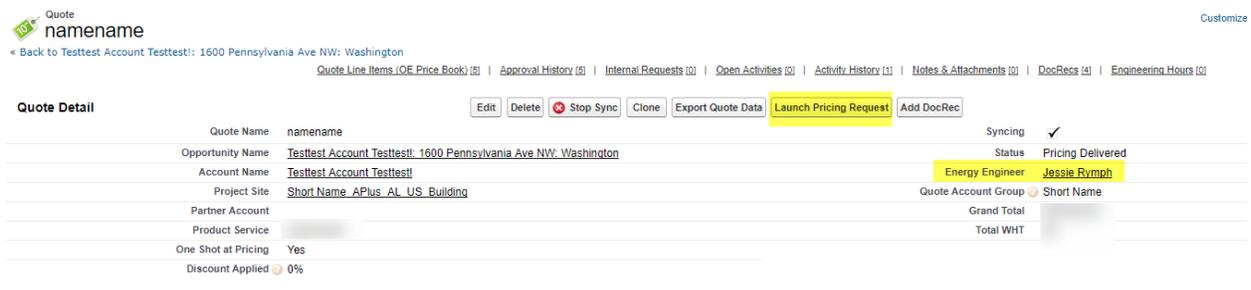
The Account Executive at Optimum Energy chooses an Energy Engineer (custom lookup to user record) on the Quote and launches the Pricing Request Flow with a button. A dynamic picklist looks for users matching certain criteria and makes a picklist of their names – that way when users change, I don't have to go back in to a hardcoded flow and adjust. The flow shows the Engineer already selected on the Quote as the default. Why am I doing this at all? Fields that are required for the Pricing Request Flow are different from fields required for the quote to save, so I walk my users through answering the right questions and fields in flow screens.

Features

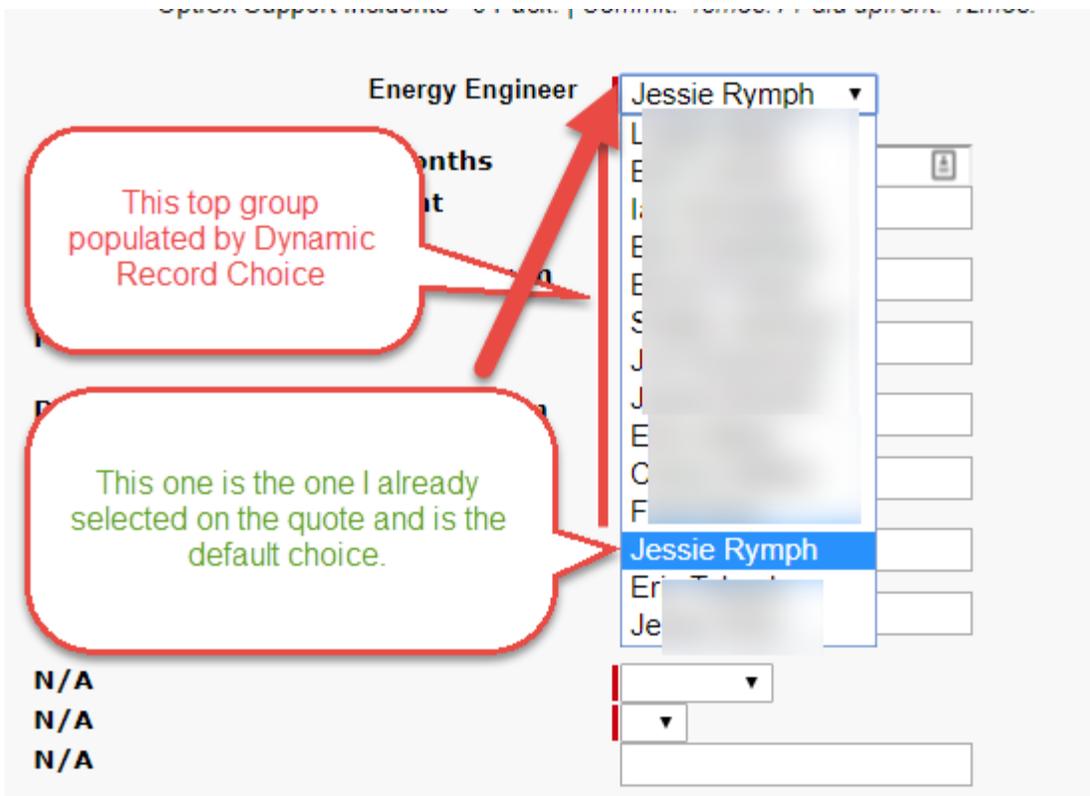
- Dynamic Record Choice – don't need to hard code the values if they may change!
- Default Picklist Value
- Screen element in the flow.

User Experience

1. User creates a quote, may or may not select the Energy Engineer on the quote, then clicks the Launch Pricing Request button.



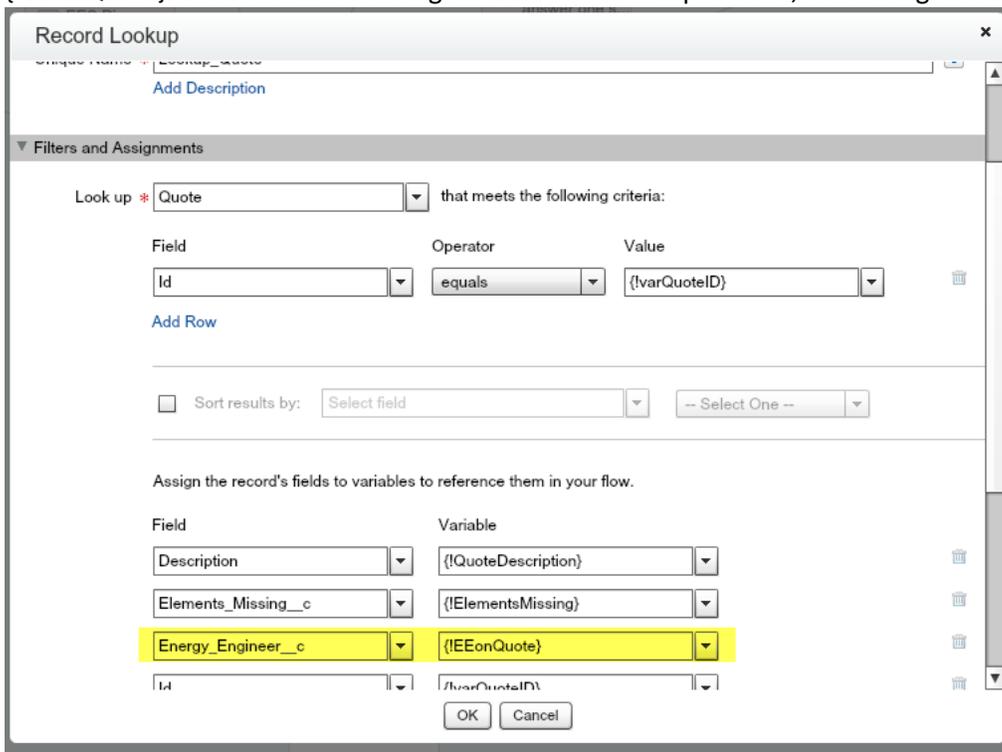
2. This launches a flow with screen elements. Among other questions, the user selects the drop down list of Energy Engineer and it's populated with a list of Energy Engineers.



Note, I'm not an energy engineer, but I list my self for testing purposes.

Steps to Build the Flow

1. Lookup to the Quote record. Assign the Energy_Engineer__c field to a text variable, {!EEonQuote}. Because this is storing the value of a lookup to User, it's storing the ID of the user.



2. Lookup to the User Record of the Energy Engineer and Assign the Name of the User to a variable.

Record Lookup [x]

▼ General Settings

Name * EnergyEngineerLookup

Unique Name * EnergyEngineerLookup [i]

[Add Description](#)

▼ Filters and Assignments

Look up * User [v] that meets the following criteria:

Field	Operator	Value
Id [v]	equals [v]	{!EEonQuote} [v]

[Add Row](#)

Sort results by: Select field [v] -- Select One -- [v]

Assign the record's fields to variables to reference them in your flow.

Field	Variable
Name [v]	{!EEFUIName} [v]

OK Cancel

3. Create a Screen element in the flow.
4. Create a field on your screen with type "Dropdown List" (um, why Salesforce, is it not called a picklist here??)

5. Add choices.

The screenshot shows the 'Screen' configuration window for 'Create Pricing Request'. The 'General Info' tab is active, showing a dropdown list for 'Energy Engineer' with a value of 'Energy_Engineer1'. The 'Choice Settings' section lists four choices: a dynamic record choice, a picklist choice with a lookup, and two static choices. Red callouts explain the dynamic record choice and the picklist choice, and note that the last two choices do not meet the criteria for a dynamic lookup.

Dynamic Record Choice

Picklist choice that is prepopulated with a lookup.

The last two choices don't meet the criteria of the dynamic lookup so I always add them like this.

- a. Create a choice for the value the user picked on the quote record.

The 'Choice' configuration window is shown, detailing the settings for a choice option. The 'General Info' section includes fields for Label, Unique Name, Value Data Type, and Stored Value, along with a checkbox for 'Show Input on Selection'.

Choice

Create an individual choice option that can be reused throughout your flow.

General Info

Label * Rich Text Edit | Clear

Unique Name * Default_Engineer

Add Description

Value Data Type Text

Stored Value {!EEonQuote}

Show Input on Selection

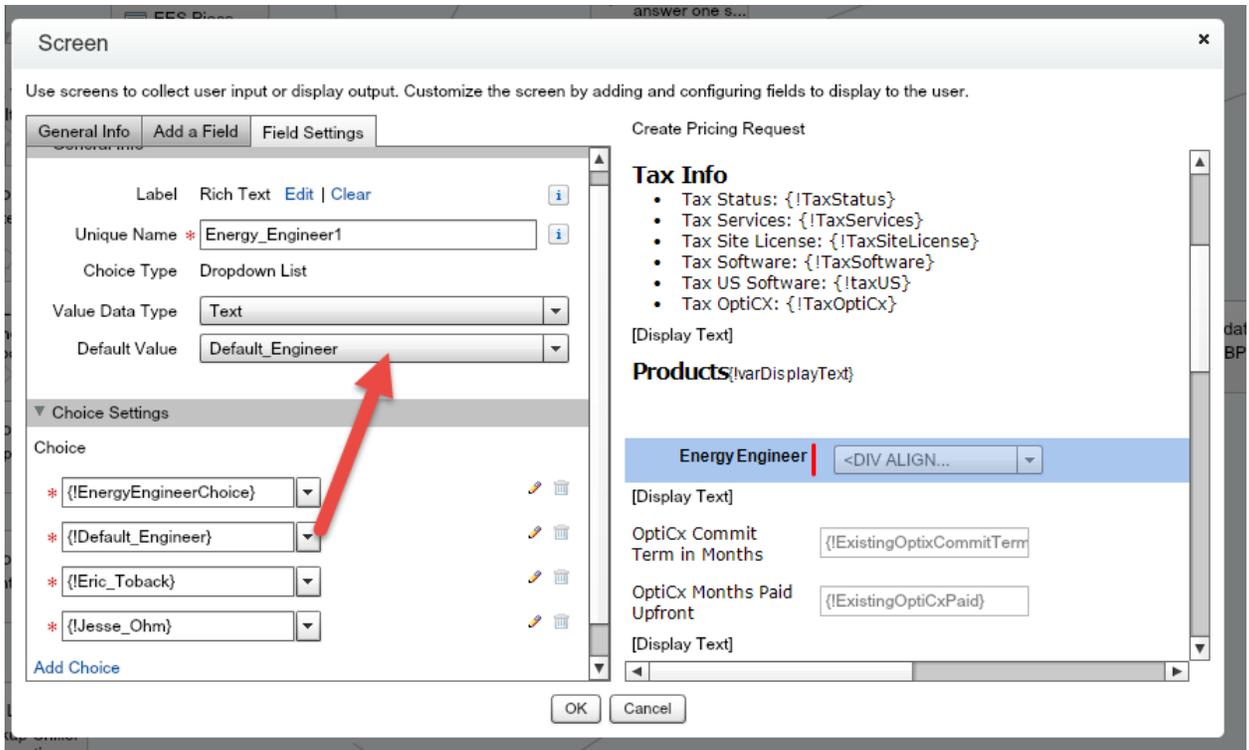
OK Cancel

Edit the Label!

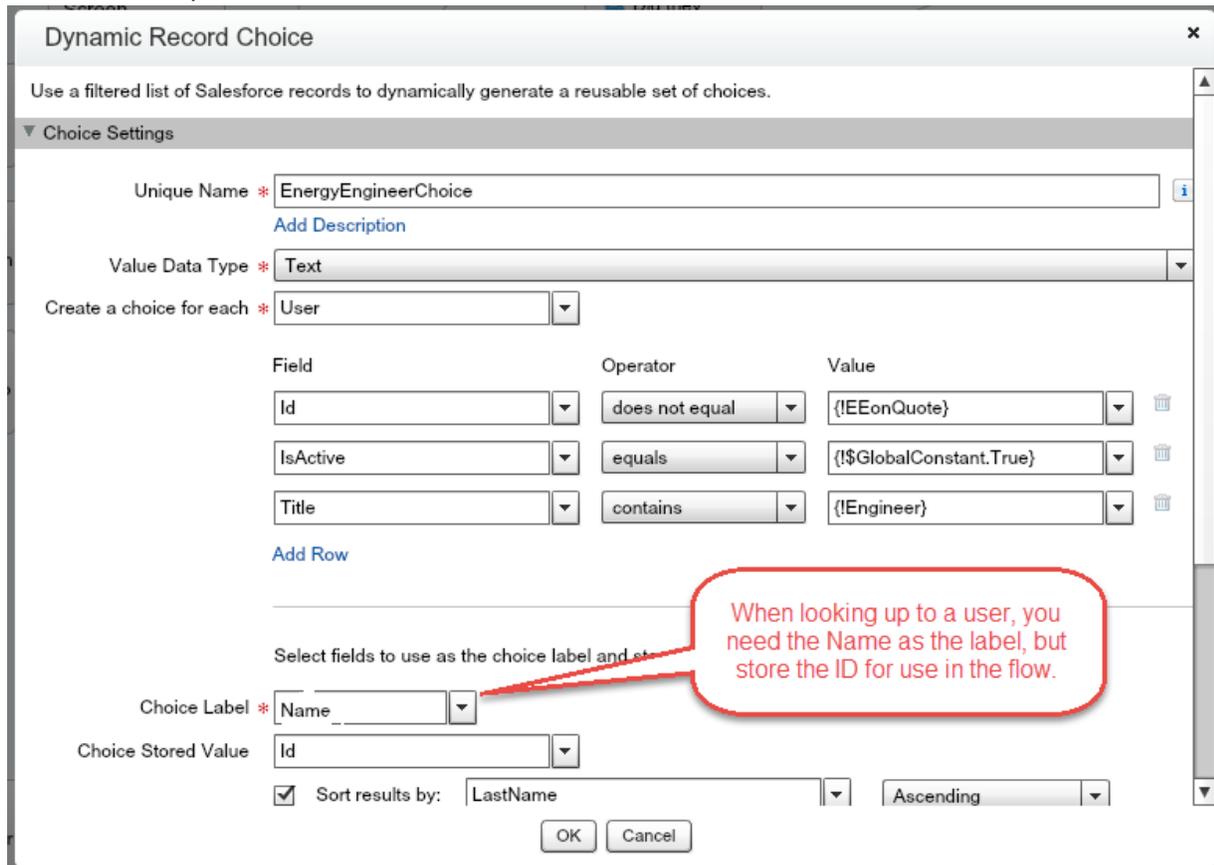
The image shows a 'Choice' configuration window with a red arrow pointing to the 'Edit' link next to the 'Label' field. The 'Edit Label' dialog is open, showing a rich text editor with a toolbar (font, size, bold, italic, underline, link, list) and a text area containing the placeholder text `{!EEFUIName}`. The 'Unique Name' field in the background is set to 'Default_Engineer'.

The stored value `{!EEonQuote}` has the ID of the user. The Label should show the user's name as the choice in the drop down list (that's the variable you looked up from the User record).

- b. Set this choice to Default value.



c. Create a new Dynamic Record Choice.



In this scenario I am looking up all active users with title contains Engineer. If the user already selected an Engineer on the quote, I do not want that value to also show up in the drop down list, because that value is set as the Default Choice (so I say ID does not equal the variable

{!EEonQuote} (the lookup to the Energy Engineer on the quote). I use the Name that as the label, while storing the ID of the user. The drop down list will then show all of the names of users who meet this criteria! I have to keep my user records up to date for the flow to change when we add new Engineers – I do not have to remember to go back in and add a name to this flow.

- d. Two of my users do not meet the criteria of the Dynamic record choice so I made stand alone choices for them. I stored their user IDs.

Choice

Create an individual choice option that can be reused throughout your flow.

General Info

Label * Eric T

Unique Name * Eric_T

Add Description

Value Data Type Text

Stored Value 00550000003Vv01

Show Input on Selection

OK Cancel

Dynamic Record Choice criteria does not allow for an “OR” option, so I couldn’t say name = this OR title equals “Energy Engineer” so I have to add these choices this way.

6. I end my flow with a Record Update of the Quote. I map the energy engineer to the “Screen Choice Field” I created earlier. This will pass the User ID of the engineer to the Quote and

update that lookup field, if necessary.

Record Update [x]

▼ Filters and Assignments

Update * Quote [v] that meet the following criteria:

Field	Operator	Value	
Id [v]	equals [v]	{!varQuoteID} [v]	[trash]

[Add Row](#)

Update record fields with variable, constant, input, or other values.

Field	Value	
Description [v]	{!Pricing_RequestNotes} [v]	[trash]
Energy_Engineer__c [v]	{!Energy_Engineer1} [v]	[trash]
Engineering_Hours_Approval__c [v]	{!EngineeringHoursExist} [v]	[trash]
InitialSubmitter__c [v]	{!CurrentUser} [v]	[trash]
Status [v]	Submitted for Approval [v]	[trash]

[Add Row](#)

[OK] [Cancel]

That's it. Please let me know if you see typos or have suggestions for improvement!