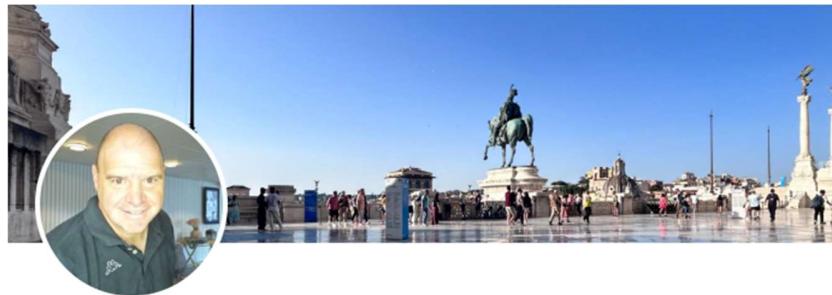


# The (Next To) Ultimate Guide to Dynamics 365 CE / Power Platform Business Process Flows

Introduction .....	3
Part 1: Understanding BPFs – Overviews .....	4
Entity in Dataverse:.....	4
The New KYC SIMPLE BPF.....	5
Here is the BPF Data Model:.....	6
Process Table .....	7
KYC SIMPLE BPF Table:.....	9
KYC SIMPLE Form .....	10
How Does This Render?.....	12
Div Header Control.....	12
Default Solution .....	14
JSON Payload.....	15
So What?.....	15
Part 2: The BPF Data model and APIs .....	16
What is in this visual editor? .....	16
Unified Process Designer.....	17
Process ID .....	17
Table Reference:.....	18
Now to BPF Process Stages .....	19
Process APP .....	21
Is There a User Case in There Somewhere? .....	22
BPF Stage History Log.....	23
Deep Dive Part 2 Conclusions .....	26
Tools .....	26
User Case, Build and Other Findings.....	27
1.) Process APP .....	27
2.) BPF History Table .....	28
3.) Canvas App .....	31
4.) In App Notification .....	34
5.) Power Automate .....	34
1a. Create KYC - Start Power Automate: .....	36
1b. Log KYC SIMPLE Stage Transition .....	41
1c. KYC SIMPLE Finished .....	47
6.) Power BI .....	50
Conclusion.....	51

## Introduction

My Name is **Patrick Kitchell**, and I have always been intrigued with using standard tools to deliver the most impactful value. I have been working with Dynamics 365 CE since v. 1.2 and Power Platform as a natural progression since the Power Platform was “birthed” out of the XRM though process that Dynamics 365 CE can be used to build anything. Please feel free to reach out and connect if this has made it into your digital hands.



**Patrick Kitchell** 

Solutions Architect | Dynamics 365 CE and Power Platform

Copenhagen Metropolitan Area · [Contact info](#)

This is a three-part series posted on my LinkedIn about Business Process Flows (BPFs). These BPFs are power tools and are used in a wide range of user cases and scenarios that can help drive transparency and understanding across an organization.

The three series is arranged as follows:

- 1.) [Understanding BPFs – Overviews](#)
- 2.) [The BPF Data model and APIs](#)
- 3.) [User Case, Build and Other Findings](#)

## Part 1: Understanding BPFs – Overviews

I have always been intrigued by these business process flows (BPF) and mainly because they are both; a nice visual if a visual is needed and they can drive process adherence automatically.

- Official documentation can be found here: [BPF](#)
- Use of Branching in BPFs: [Branching](#)



The basics of BPFs are that for **each new BPF** a new entity is created in Dataverse. The BPF instance is related to a primary entity or to the entity where it is meant to start the visualization of the process.

### Entity in Dataverse:

I have created a new BPF around Know Your Customer just to highlight an example. The new KYC Simple BPF table is below:

Contact	Active Stage	Status	Created On	Process
Patrick Kitchell	KYC STAGE 1	Active	6/10/2025 2:14 PM	KYC Simple
Select lookup	Select lookup			Select lookup

## The New KYC SIMPLE BPF

Data in the BPF Entity is collected as data in the KYC Simple entity

Contact	Active Stage	Status	Created On	Process
Patrick Kitchell	KYC STAGE 1	Active	6/10/2025 2:14 PM	KYC Simple
Select lookup	Select lookup			Select lookup

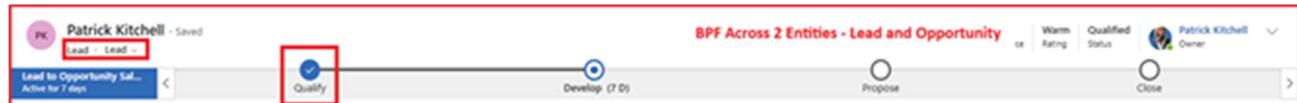
Here are 2 examples of BPFs:

Example 1: BPF on a Contact – 3 Stages



Example 2: BPF Across Lead and Opportunities – 4 Stages

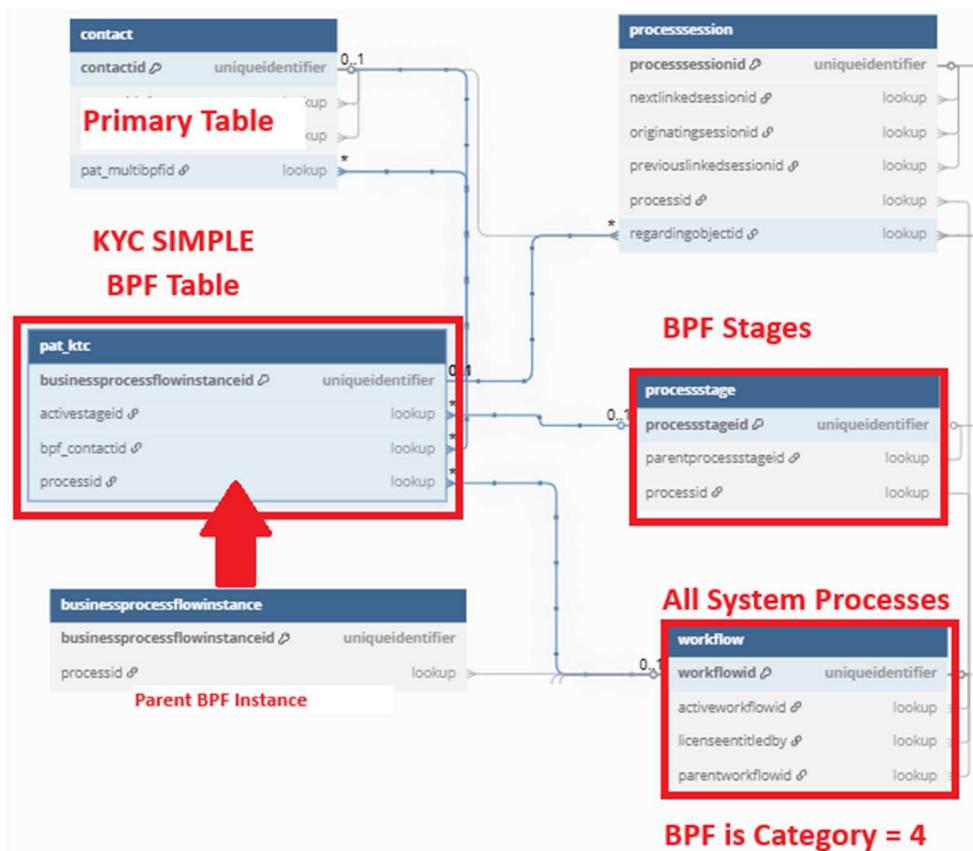
Start with 1 stage on the Lead



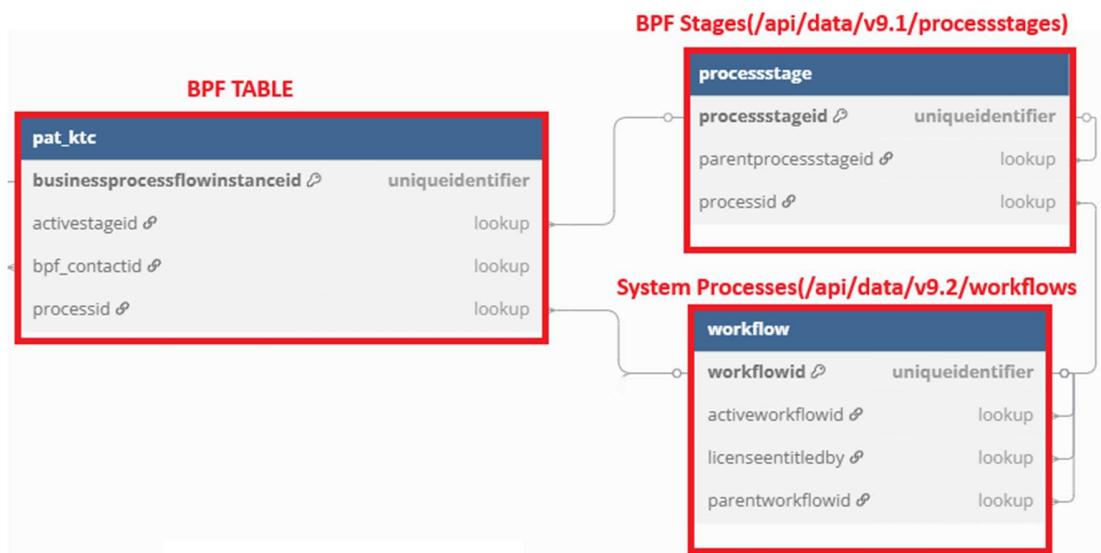
Finish with 3 stages on an Opportunity



## Here is the BPF Data Model:



The data model for BPFs using my KYC Simple BPF creates a **Business Process Flow Instance**. This instance is related to sessions of this process (Process Sessions) and each stage in the BPF is created in a table called Process Stages. The process sessions are a well-known table that holds all the process sessions for various types of automations in both Dynamics 365 CE and Power Platform



## Process Table

The Process Table holds a wide range of information about these system processes. System Processes are a range of automations like Classic Workflows, Actions, Flows, BPFs.

The screenshot shows the Dynamics 365 Process App interface. The left sidebar lists various entities: Home, Recent, Pinned, Customers, Accounts, Contacts, Sales, Leads, Opportunities, KYCSimple, and Processes. The Processes item is selected. The main area displays a list titled "Activated Processes" with the following columns: Process Name, Category, and Primary Entity. The listed items include:

Process Name	Category	Primary Entity
(DEPRECATED) BatchAddMarketingListMembersMasterWork...	Workflow	Marketing List
ActivateProcesses	Action	none
ActivateSdkMessageProcessingSteps	Action	none
ActivityPropagation	Workflow	Quick Campaign
AddBotUser	Action	none
After Meeting	Business Process Flow	Appointment
Agreement Business Process	Business Process Flow	Agreement

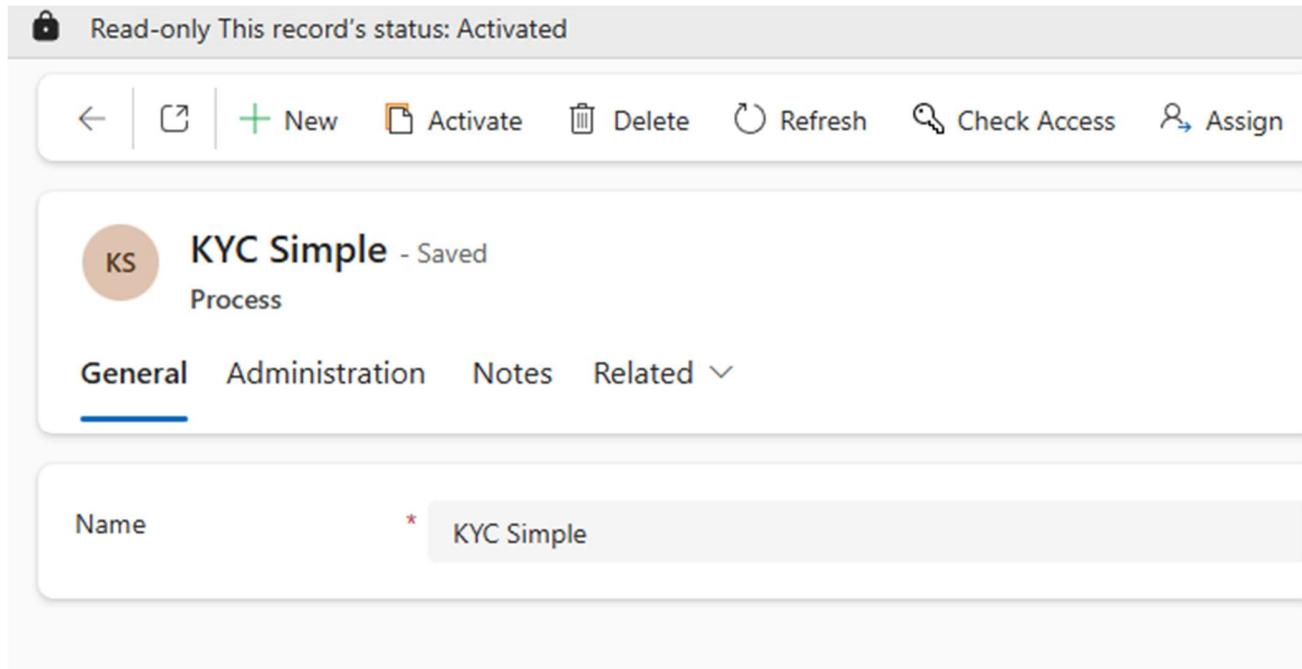
Take for example the **KYC Simple BPF** – it is also located in the Process Table here and it possible to see all BPFs in the Process Table by adding the Process Table to an app (as shown above) and then filtering the category on “**Business Process Flow**”.

The screenshot shows a list titled "My Processes" with the following columns: Process Name, Category, Primary Entity, Status, Mode, Modern Flow Type, Scope, and ProcessTriggerS... . The listed item is:

Process Name	Category	Primary Entity	Status	Mode	Modern Flow Type	Scope	ProcessTriggerS...
KYC Simple	Business Process Flow	Contact	Activated	Background	PowerAutomateFlow	Organization	Form

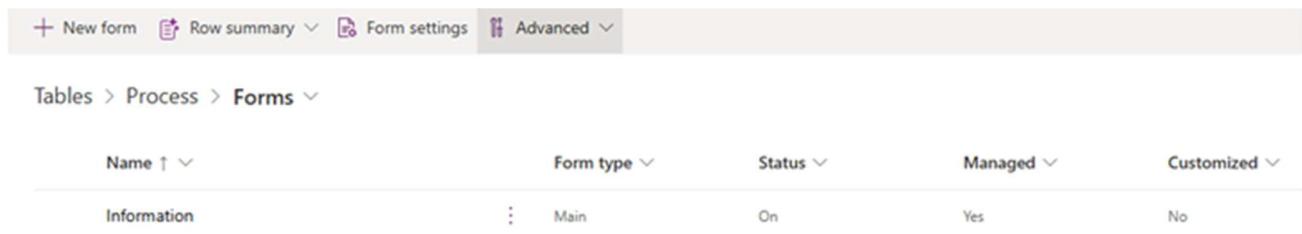
To be clear, when clicking on the BPF in the list data(above) an information form will open for each BPF, and this information form **can't be edited**. It is not possible to create a new shape for this table.

## KYC Simple Information Opened



The screenshot shows the Dynamics 365 Business Central interface for a process named "KYC Simple". The top navigation bar includes standard actions like New, Activate, Delete, Refresh, Check Access, and Assign. The main area displays the process details under the "General" tab, which includes the process name "KYC Simple" and its status as "Saved". Other tabs available are Administration, Notes, and Related.

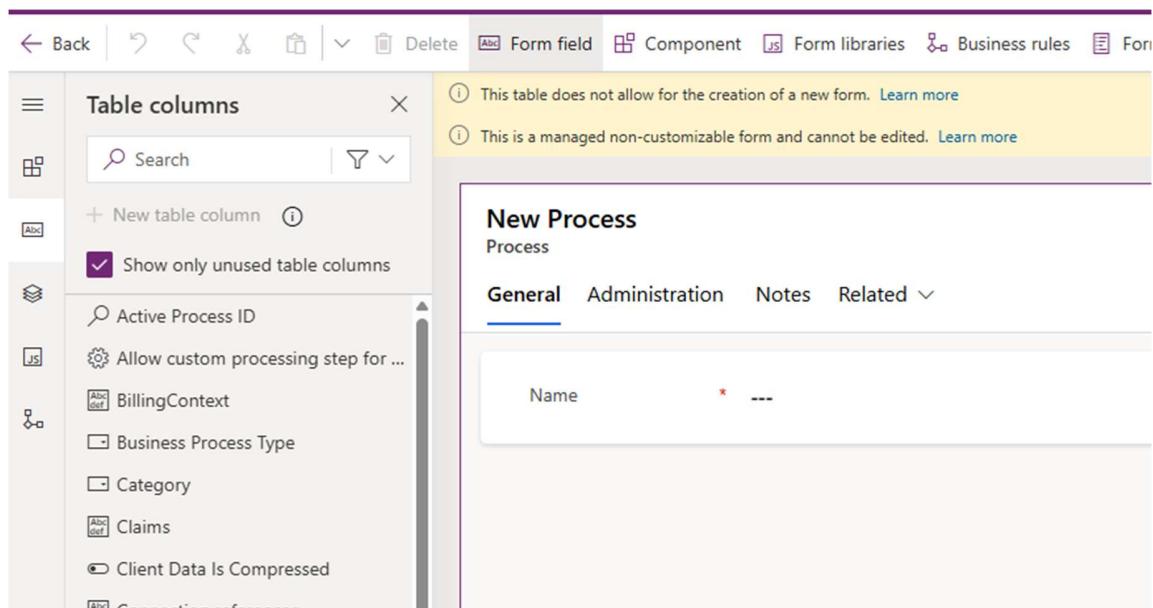
When looking at the Process Table and Forms this is what is shown.



The screenshot shows the Dynamics 365 Business Central Process Table under the "Forms" section. The table lists one form named "Information". The columns include Name, Form type, Status, Managed, and Customized. The "Information" form is listed as Main, On, Yes, and No respectively.

Name ↑ \	Form type \	Status \	Managed \	Customized \
Information	Main	On	Yes	No

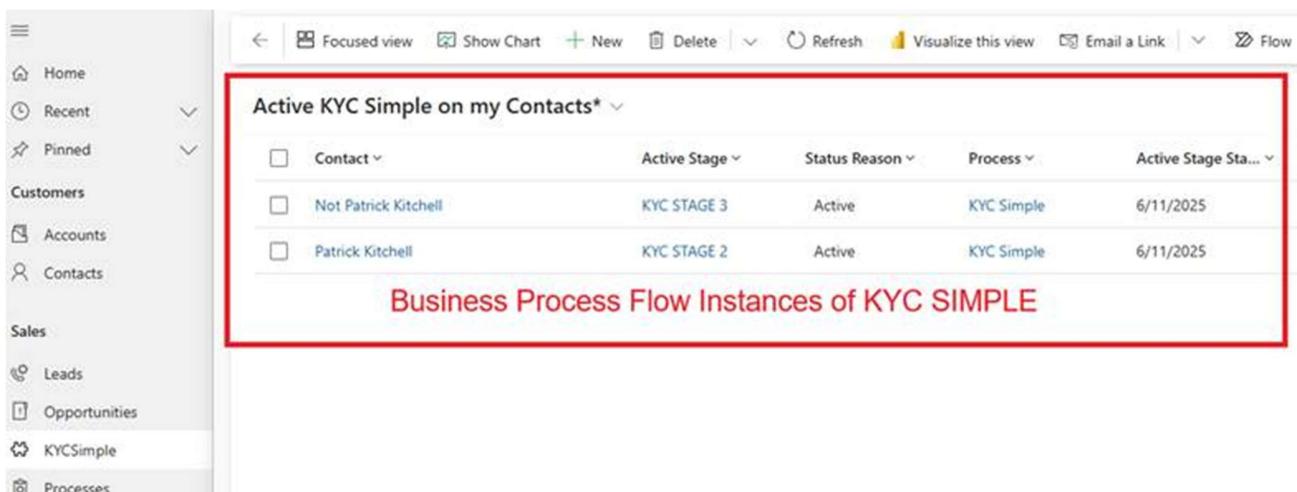
When trying to create a new form the system will not allow it. 



This screenshot shows a 'New Process' form in Dynamics 365. A yellow warning bar at the top indicates that the table does not allow for the creation of a new form and that it is a managed non-customizable form. The 'General' tab is selected, showing a 'Name' field.

## KYC SIMPLE BPF Table:

As mentioned above with the Process Table it is also possible to add the BPF Tables to an application. As shown below the KYC Simple Table is added to my app and from the navigation it shows each KYC Simple business process instance created on the primary entity. The process instance is for creation of the BPF on the primary contact.

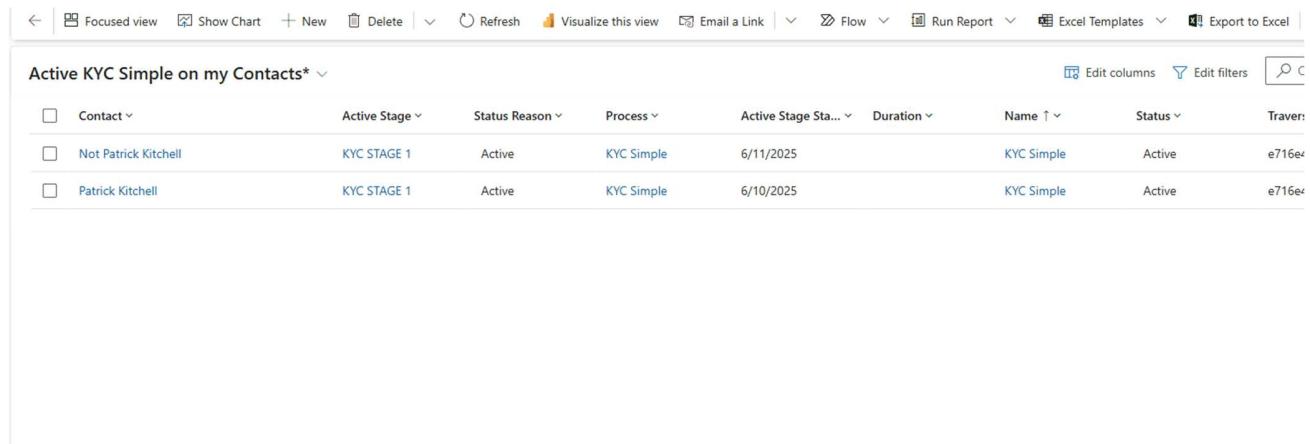


Contact	Active Stage	Status Reason	Process	Active Stage Start Date
Not Patrick Kitchell	KYC STAGE 3	Active	KYC Simple	6/11/2025
Patrick Kitchell	KYC STAGE 2	Active	KYC Simple	6/11/2025

**Business Process Flow Instances of KYC SIMPLE**

## Updating BPF Stages in Excel Online 😊

**Like any common Table in Power Platform** they can open these in excel online, edited and sent back for an update (system import). As a just as an example and move the ACTIVE STAGE (Stages will be discussed in another post):

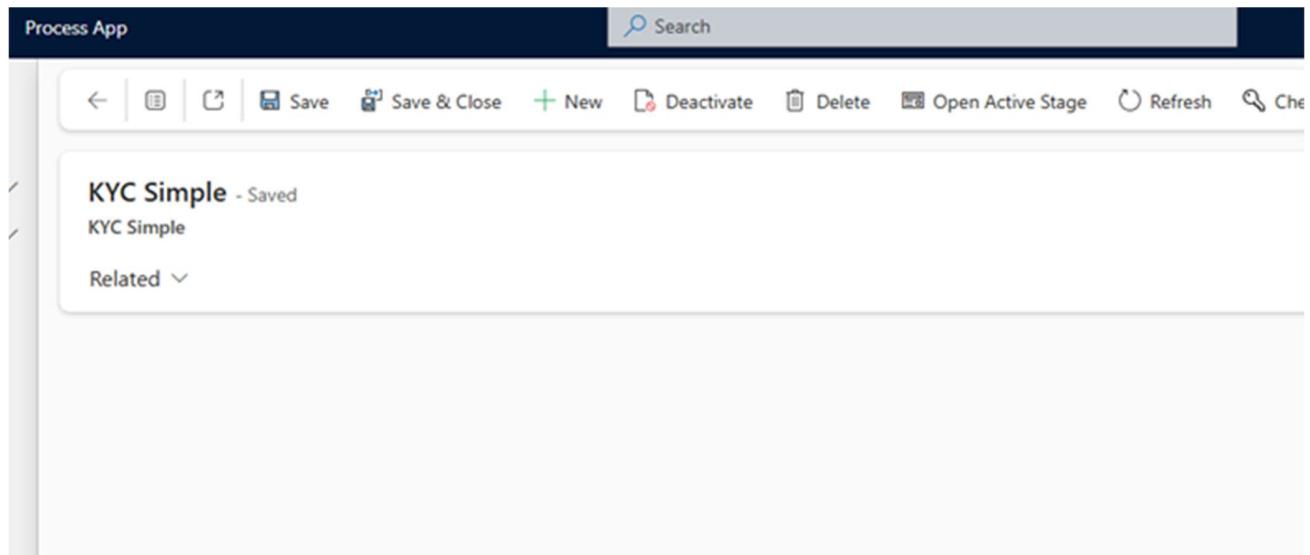


The screenshot shows an Excel Online table titled "Active KYC Simple on my Contacts\*". The table has columns: Contact, Active Stage, Status Reason, Process, Active Stage Sta..., Duration, Name, Status, and Traver:. There are two rows of data:

Contact	Active Stage	Status Reason	Process	Active Stage Sta...	Duration	Name	Status	Traver:
Not Patrick Kitchell	KYC STAGE 1	Active	KYC Simple	6/11/2025		KYC Simple	Active	e716e4
Patrick Kitchell	KYC STAGE 1	Active	KYC Simple	6/10/2025		KYC Simple	Active	e716e4

## KYC SIMPLE Form

AS for the BPF form it uses the Process Table information form that can't be edited, and it is not possible to create a new form for this table:

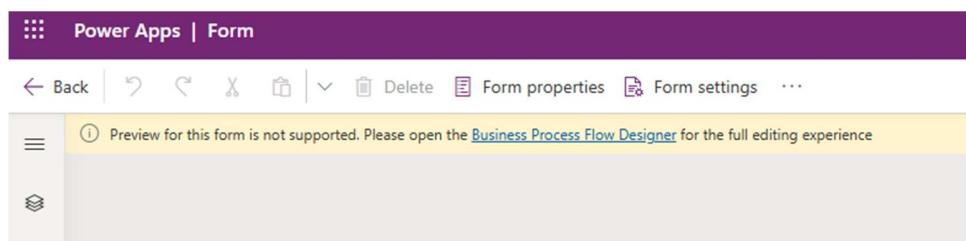


## KYC SIMPLE BPF > FORM

Name ↑	Form type	Status	Managed
Information	Main	On	No

### KYC Simple BPF Information Form

The difference between this Table and Process Table is it doesn't even render the form for opening or editing. This is because the editing of the BPF takes place in the Business Process Flow Designer

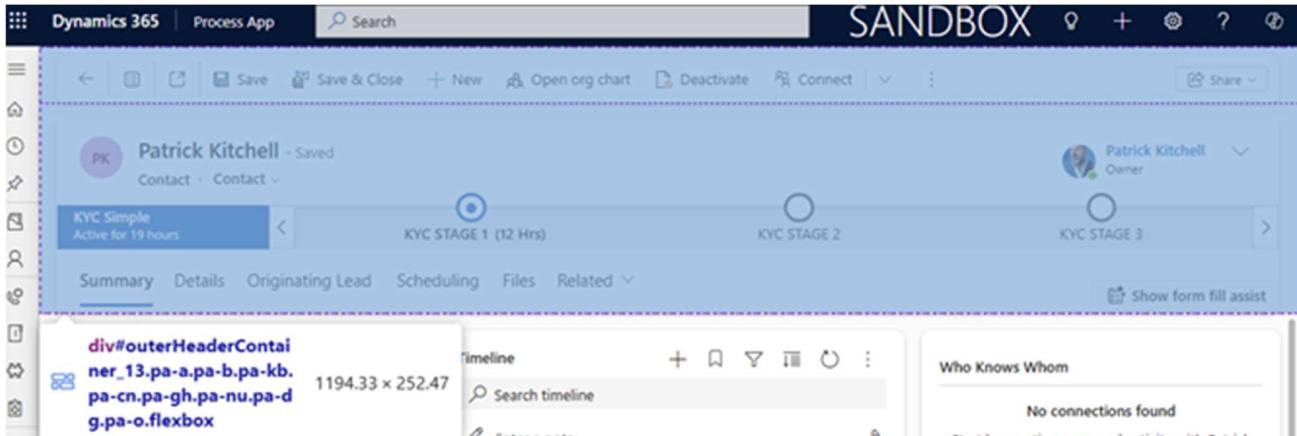


Clicking on the **Business Process Flow Designer** text above will open the designer and this a special page combining the BPF Table, Process Stage Table into 1 experience.

## How Does This Render?

I came to realize that I do not understand how this works because it is an entity embedded in the header of another entity. BPF Entity in the header of a contact entity?

In the Dev Tools in chrome, it looks like this:



## Div Header Control

And when digging into just the BPF – than we see it is a control embedded via a <div>

```
<div role="presentation" class="pa-a pa-ae pa-cn pa-ae1 flexbox"></div>
</div>
<div id="processControlBreadCrumb_13" role="presentation" class="pa-a pa-ae flexbox">flex == $0
  <div id="hncContainer" role="presentation" class="pa-cz pa-cn pa-ao pa-dl pa-ph pa-nc flexbox">flex
    <div id="pa-cn" role="presentation" class="pa-cn flexbox" data-lp-id="MscrmControls.Containers.ProcessBreadCrumb|MscrmControls.Containers.ProcessBreadCrumb|contact">flex
      <div id="MscrmControls.Containers.ProcessBreadCrumb-processControlContainer" data-id="MscrmControls.Containers.ProcessBreadCrumb-processControlContainer" role="presentation" class="pa-db pa-cn flexbox">flex
        <div id="MscrmControls.Containers.ProcessBreadCrumb-processHeaderContainer" data-id="MscrmControls.Containers.ProcessBreadCrumb-processHeaderContainer" role="presentation" class="pa-x pa-a pa-db pa-cn pa-i flexbox">flex
          <div id="MscrmControls.Containers.ProcessBreadCrumb-processHeaderDataContainer" aria-live="assertive" role="status" data-id="MscrmControls.Containers.ProcessBreadCrumb-processHeaderDataContainer" class="pa-n pa-fn pa-b pa-ajc pa-ajd pa-aje pa-kq pa-id pa-ih pa-ab pa-alz pa-f flexbox">...</div> (flex)
          <div id="MscrmControls.Containers.ProcessBreadCrumb-processHeaderMobileRightSectionContainer13" data-id="MscrmControls.Containers.ProcessBreadCrumb-processHeaderMobileRightSectionContainer13" role="presentation" class="pa-alz flexbox">...</div> (flex)
        </div>
        <div id="MscrmControls.Containers.ProcessBreadCrumb-processControlAlertContainer" aria-live="assertive" role="alert" data-id="MscrmControls.Containers.ProcessBreadCrumb-processControlAlertContainer" class="pa-bf pa-de pa-j pa-ala "></div>
```

## A Process Bread Crumb HeaderContainer

In the Default solution the control for the ProcessBreadCrumb control looks like this:

Objects < + New Publish all customizations ...

Default Solution > Custom controls

Display name ↑	Name	Type	Managed
Intelligence.FormProcessorControl.FormProcessor	Intelligence.FormProcesso...	Custom Control	Yes
Intelligence.ReceiptProcessorControl.ReceiptProcessor	Intelligence.ReceiptProces...	Custom Control	Yes
MscrmControls.Containers.ProcessBreadCrumb	MscrmControls.Container...	Custom Control	Yes
MscrmControls.Containers.ProcessStageControl	MscrmControls.Container...	Custom Control	Yes
MscrmControls.Containers.SwitchProcessGridControl	MscrmControls.Container...	Custom Control	Yes

## Default Solution

### Opening the Control Properties:

In the properties of the control, it is possible to see the various JS libraries and it states "IsVirtual" = True, which is interesting.

The screenshot shows the Dynamics 365 Properties pane. The left sidebar lists 'Default Solution > Custom controls > MscrmControls.Containers.ProcessBreadCrumb > Solution L'. The main pane has tabs 'Updated' and 'All'. A table lists properties and their values. The 'clientjson' property is expanded, showing its JSON value. Other properties listed include 'compatibledatatypes', 'componentstate' (Published), 'createdby' (64234b47-be87-462e-bc47-d4edfb676917), and 'creationon' (4/26/2019 7:07 AM).

Property name	Value
clientjson	<pre>["CustomControlId": "e7fe1ee9-6d9f-4516-a5a2-69f5a1e4337e", "Name": "MscrmControls.Containers.ProcessBreadCrumb", "ConstructorName": "ProcessBreadCrumb", "ControlMode": "Both", "DisplayName": "CC_Process_Control_Name", "Namespace": "MscrmControls.Containers", "IsVirtual": true, "Properties": {"Resources": [{"Name": "mscommon", "LoadOrder": 1, "LibraryName": "mscommon", "Type": 0}, {"Name": "processcontrolmanager", "LoadOrder": 2, "LibraryName": "processcontrolmanager", "Type": 0}, {"Name": "MscrmControls.Containers.ProcessBreadCrumb_ProcessBreadCrumb.js", "LoadOrder": 4, "Type": 4}, {"Name": "string/ProcessControlResources.1033.res", "LoadOrder": 0, "Type": 5}], "DataSets": [], "Properties": [{"Name": "value", "Type": "BusinessProcessFlow", "Hidden": false, "Required": true, "Usage": 0}], "FeatureUsage": [], "PlatformAction": [], "Events": [], "Actions": []}, "VersionNumber": 0, "OverallVersionNumber": 0, "MappingFileName": "design-maps/map.xml", "FullName": "MscrmControls.Containers.ProcessBreadCrumb", "Children": [], "ControlManifestVersion": "1.1.142"}]</pre>
compatibledatatypes	
componentstate	Published
createdby	64234b47-be87-462e-bc47-d4edfb676917
creationon	4/26/2019 7:07 AM

## JSON Payload

### The Manifest is here:

The Manifest gives some information also.

manifest	<manifest> <control namespace="MscrmControls.Containers" constructor="ProcessBreadcrumb" version="1.1.142" display-name- key="CC_Process_Control_Name" control- type="virtual" hidden="true" design-mapping- file="design-maps/map.xml"> <modes> <read /> <edit /> </modes> <property name="value" required="true" display-name- key="CC_Process_Value" description- key="CC_Process_Value_Desc" usage="bound" of-type="BusinessProcessFlow" /> <resources> <library name="mscommon" version="&gt;=1" order="1"> <packaged_library path="libs/mscommonlib- 1.0.0.js" version="1.2.20" /> </library> <library name="processcontrolmanager" version="&gt;=1" order="2"> <packaged_library path="libs/processcontrolmanager-1.0.0.js" version="1.0.98" /> </library> <resx path="strings/ProcessControlResources.1033.r esx" version="1.0.3" /> <code path="ProcessBreadcrumb.js" order="4" /> </resources> </control> </manifest>
----------	---

## So What?

Like everything that looks simple under the hood it is not. There is a lot more information to dive into with the business process controls that I will provide in another deep dives especially how to use it in very practical ways like

- How does the BPF Keep Time on the process and in each Stage?
- How do stages work on a BPF?
- Moving the BPF stages automagically.
- Having more than one PBF on an entity and why?
- Visualizing multiple PBFs on an Entity.
- Importing and exporting
- Power BI and BPFS
- Other tips and tricks

## Part 2: The BPF Data model and APIs

The last Deep Dive we ended here -- **a visual editor for BPFs.**

### What is in this visual editor?

To better understand the visual designer than it is best to understand the data model for BPFs because the visual designer is combining multiple tables (and relationships) into it.

The screenshot shows the Microsoft Power Apps Unified Process Designer interface. At the top, there are navigation buttons (Back, Forward, Home, Refresh) and a URL bar showing the path: /Tools/ProcessControl/UnifiedProcessDesigner.aspx?id=%7b13B7C019-EF45-F011-8779-7C1E5236AC0A%7d#. Below the header is a toolbar with icons for Add, Cut, Copy, Paste, Delete, Snapshot, and Connector. The main area is divided into sections:

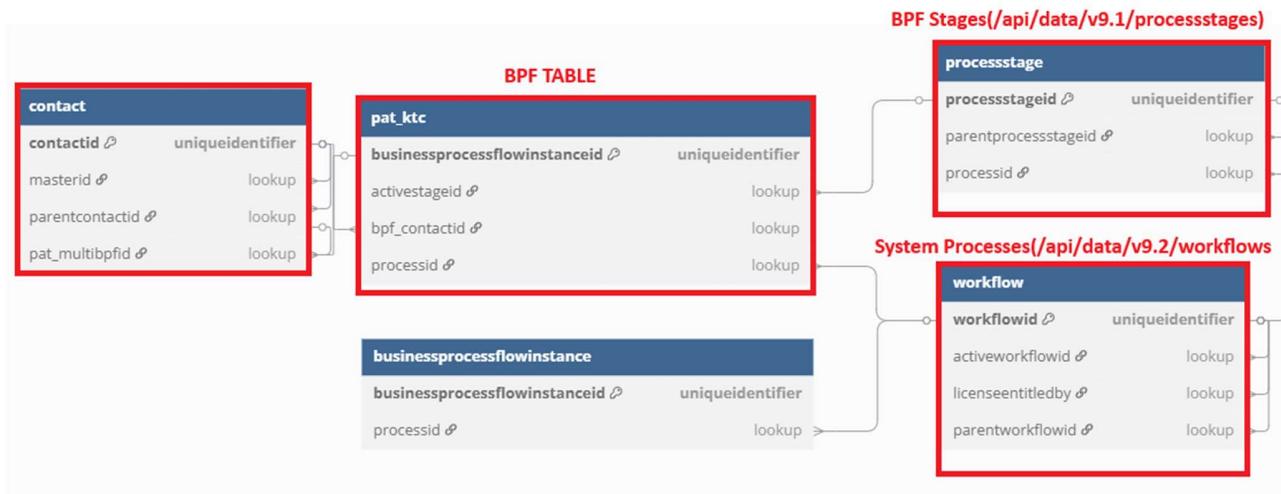
- KYC Simple**: A form with fields for Process Name (KYC Simple), Owner (Patrick Kitchell), Name (pat\_ktc), and Description (empty). Primary Entity is set to Contact and Category is Business Process Flow.
- Process Stages**: A diagram showing three stages connected sequentially:
  - KYC STAGE 1**: Contact (1 record, 0 steps)
  - KYC STAGE 2**: Contact (1 record, 0 steps)
  - KYC STAGE 3**: Contact (1 record, 0 steps)

To better understand the visual designer than it is best to understand the data model for BPFs because the visual designer is combining multiple tables (and relationships) into it.

When we look at the **Unified Process Designer** above it contains the following tables from the data model.

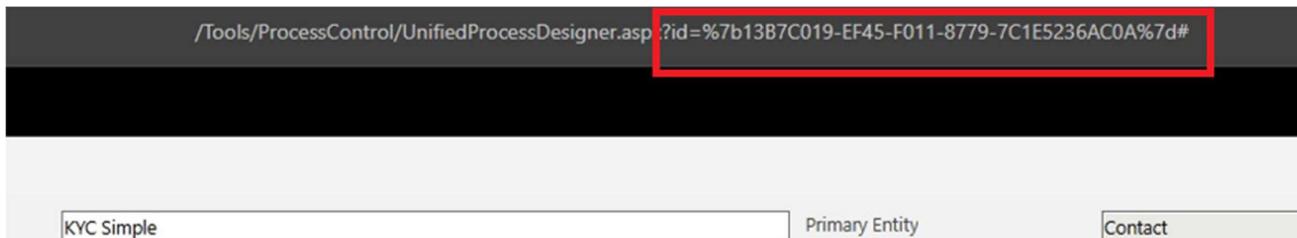
1. **A Primary** related Table = KYC Simple starts on a **Contact**
2. **The KYC Simple BPF** is a Table Itself
3. **A BPF Stages** belong to another table called Process Stages
4. **Workflows (Processes)** are the table for all System Workflows where all BPFs are of Category = 4

Remember the data model from above – here is a refresher:



## Unified Process Designer

When we look at the Unified Process Designer URL it is pointing at: **id:13b7c019-ef45-f011-8779-7c1e5236ac0a** which is the process ID used for the BPF Workflow (**WorkflowID**) and in the Process Stage(**ProcessID\_Value**)



## Process ID

### Workflows (Processes)

I think that most people that have worked with Dynamics 365 CE / PowerApps know about workflows(processes) and how this all works. It should be noted that the table schema/logical name is "**Workflow**" and the display name is "**Processes**"

The screenshot shows the Dynamics 365 Process App interface. The left sidebar contains navigation links such as Home, Recent, Pinned, Customers, Contacts, Sales Activity Social ..., Sales, Leads, Opportunities, Processes, BPF Stage Histories, KYCSimple, Customer Journey, and Onboarding Journey. The main area is titled "My Processes" and displays a table with the following data:

<input type="checkbox"/> Process Name ↑	Category	Primary Entity
<input type="checkbox"/> Customer Journey	Business Process Flow	Contact
<input type="checkbox"/> KYC Simple	Business Process Flow	Contact
<input type="checkbox"/> Onboarding Journey	Business Process Flow	Contact
<input type="checkbox"/> Purchase Order Business Process	Business Process Flow	Purchase Order
<input type="checkbox"/> Work Order Business Process	Business Process Flow	Work Order

## Table Reference:

I just filtered out the processes on category = "Business Process Flow" in the standard UI view builder.

Using the **API**, It is possible to see your processes via the web where workflow.category = 4 means BPF.

**Other values** include: 0 = Definition | 1 = Activation | 2 = Action | 3 = Business Rule | 4 = Business Process Flow

## All Workflows(process)

[environment]/api/data/v9.2/workflows

## Or just the active BPFs

[environment]/api/data/v9.1/EntityDefinitions?\$select=SchemaName,LogicalName,DisplayName&\$filter=IsBPFEntity%20eq%20true

[environment]/api/data/v9.1/EntityDefinitions?\$select=SchemaName,LogicalName,DisplayName&\$filter=IsBPFEntity%20eq%20true

## Or a single BPF

[environment]/api/data/v9.1/[BPF Schema Name]

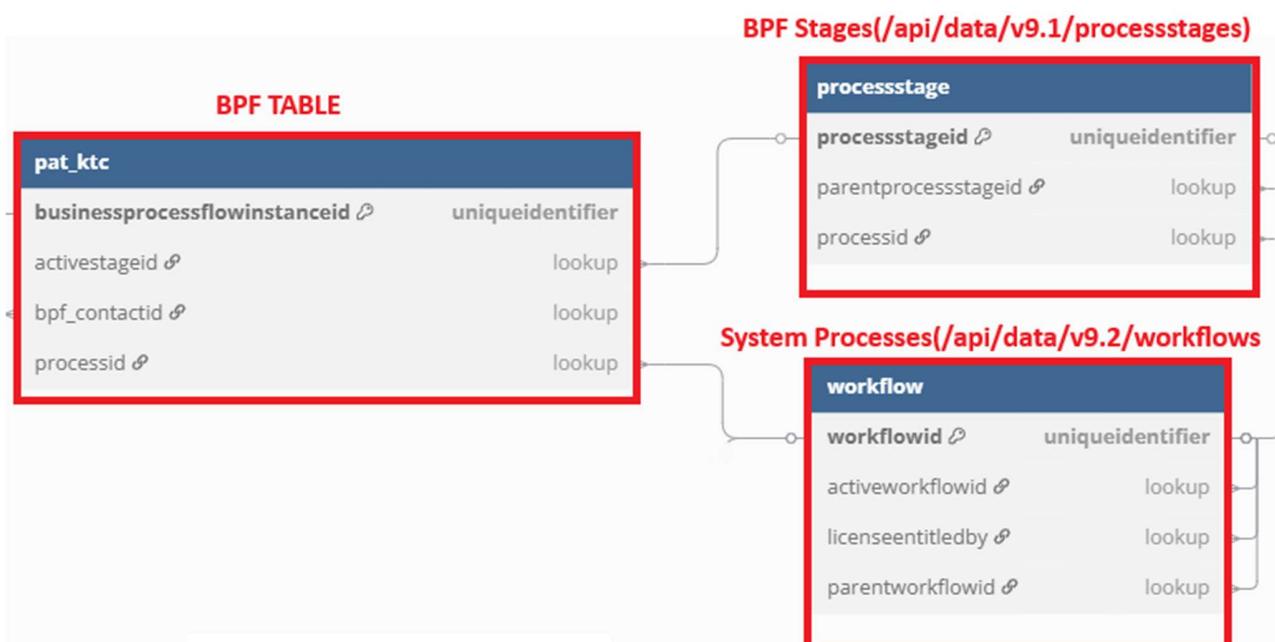
## Now to BPF Process Stages

For each Process Stage added to the BPF in the visual editor:



## Process Stages

A New Process Stage is created in the **Process Stage Table**. To find the process stages just use the API ***because this is hidden and can't be added to the UI of any model driven app.***



## API

Using the API makes it easy find the Process Stages and their Stage Ids

[your env]/api/data/v9.1/processstages

## Process Stages ID

Can be found in through the API and searching for the name of the Process. I used "KYC" and it returnend the process stages to me :

```
"@odata.etag": "W/\\"9236990\\",
```

```
"operationkind": null,  
"parametervalue": null,  
"_processid_value": "13b7c019-ef45-f011-8779-7c1e5236ac0a",  
"operationtype": null,  
"clientdata": "[{\\"DisplayName\\":\\"Email\\",\\"DisplayLabelId\\":\\"47feffa6-475f-412b-86ed-3f2361f2c47e\\",\\"Type\\":\\"Field\\",\\"Field\\":{\\\"AttributeName\\":\\"emailaddress1\\",\\"IsRequired\\":true}}, {\\"DisplayName\\":\\"First Name\\",\\"DisplayLabelId\\":\\"2fd4dff4-f6da-4bcc-80bd-eb69ca7210d5\\",\\"Type\\":\\"Field\\",\\"Field\\":{\\\"AttributeName\\":\\"donotbulkpostalmail\\",\\"IsRequired\\":true}}, {\\"DisplayName\\":\\"Birthday\\",\\"DisplayLabelId\\":\\"72e0d289-d13b-4fe3-9c9e-5beeb220e8ec\\",\\"Type\\":\\"Field\\",\\"Field\\":{\\\"AttributeName\\":\\"birthdate\\",\\"IsRequired\\":true}}]}"  
,  
"owningbusinessunit": "c0e7ee9b-5822-f011-8c4e-7c1e5276d9d7",  
"stagename": "KYC STAGE 1",  
"_parentprocessstageid_value": null,  
"versionnumber": 9236990,  
"istrigger": false,  
"stagecategory": -1,  
"operationid": null,  
"parametername": null,  
"primaryentitytypecode": "contact",  
"processstageid": "e716e4e9-c2eb-475d-a379-31c63efedc80",  
"_ownerid_value": "d0eeee9b-5822-f011-8c4e-7c1e5276d9d7",  
"connector": null  
}
```

The other way to get them is to add the BPF to your app and you can create a view of the BPF. See how I added the BPF called KYC SIMPLE to the UI.

The screenshot shows the Dynamics 365 Process App interface. On the left, a navigation bar lists various entities: Home, Recent, Pinned, Customers, Contacts, Sales Activity Social ..., Sales, Leads, Opportunities, Processes, BPF Stage Histories, KYCSimple (which is highlighted with a red box), Customer Journey, and Onboarding Journey. In the center, a list view titled "Active KYC Simple on my Contacts\*" is displayed. The list contains three items, each with a checkbox, a name, and a GUID. The first item is "Process" with ID e716e4e9-c2eb-475d-a379-31c63. The second and third items are both "KYC Simple" with the same ID e716e4e9-c2eb-475d-a379-31c63.

	Process	Traversed Path
<input type="checkbox"/>	KYC Simple	e716e4e9-c2eb-475d-a379-31c63
<input type="checkbox"/>	KYC Simple	e716e4e9-c2eb-475d-a379-31c63

## Process APP

**The table KYC Simple** can open but it is not possible to add any fields to a form or create a new form as the entity is locked down. As mentioned before the form is locked and limited.

The screenshot shows the LinkedIn KYC Simple view. At the top, there is a toolbar with icons for back, forward, save, new, deactivate, delete, and open active stage. Below the toolbar, the title 'KYC Simple - Saved' is displayed, followed by the label 'KYC Simple'. A 'Related' dropdown menu is visible. The main content area is currently empty.

## Process Stages ID in the KYC Simple View

The **Process Stage IDs** are found in the view, and they are called **Transversed Path**. The Transversed path shows which Stages the BPF is on or has been on by listing the ID(s) and separated by a comma.

What is important to note is that the KYC Simple BPF Instance only tracks the stages via the transversed path. It stamps the time in each stage and it overwrites the data regarding the stages.

Active KYC Simple on my Contacts*			BPF Stages(Process Stages Table Unique ID)
Contact	Status Reason	Process	
<input type="checkbox"/> Micky Mouse	Active	KYC Simple	e716e4e9-c2eb-475d-a379-31c63efedc80,e7e2acd-09d6-45e3-8a9c-a21320774b69,89a593a9-52fc-4d61-80fa-0dd43dc63a08
<input type="checkbox"/> A Goofy Mule	Active	KYC Simple	e716e4e9-c2eb-475d-a379-31c63efedc80

**KYC Stage 1**

## Is There a User Case in There Somewhere?

The user case wants to run simultaneously difference Business Process Flows (BPF) on a customer.

In the User Case I am in banking:

- The Bank needs to follow multiple processes/procedures simultaneously and trigger stages programmatically.
- It is not mandatory that single BPFs to be visual on the primary table(contact) because multiple BPFs are in the background.
- Finally, they want to be able to track where the contact is in these various processes and progress should be visible to the people that need to see them.

## ThreeBusiness Processes

For this user case there are - 3 BPFs - with 3 stages each:

### 1. Know Your Customer (KYC is shown on contact)



### 2. Onboarding Journey



### 3. Customer Journey



## BPF Stage History Log

Now there is a user case with the ask to have multiple BPFs on a single table and to be able to track BPFs. A custom Table is created a **BPF Stage History Log** to track the creation, stage changes and process finish for the BPF.

### Active BPF Stage Histories

<input type="checkbox"/> Contact	BPF Instance	BPF Record ID	Previous Stage N...	Multi BPF Lookup
<input type="checkbox"/> Not Patrick Kitchell	KYC Simple	d6f97467-ae46-f0...		KYC Simple
<input type="checkbox"/> Patrick Kitchell	KYC Simple	a193a186-f445-f0...	KYC STAGE 3	KYC Simple
<input type="checkbox"/> Patrick Kitchell	Onboarding J...	9a81f01e-5f48-f0...	Onboarding Stage 1	Onboarding Jour...
<input type="checkbox"/> Not Patrick Kitchell	Customer Jo...	2e5397b0-6048-f...	Customer Journey S...	Customer Journey
<input type="checkbox"/> Patrick Kitchell	Onboarding J...	9a81f01e-5f48-f0...		Onboarding Jour...
<input type="checkbox"/> Patrick Kitchell	Onboarding J...	9a81f01e-5f48-f0...	Onboarding Stage 2	Onboarding Jour...
<input type="checkbox"/> Patrick Kitchell	Onboarding J...	9a81f01e-5f48-f0...	Onboarding Stage 3	Onboarding Jour...
<input type="checkbox"/> Patrick Kitchell	Onboarding J...	9a81f01e-5f48-f0...	Onboarding Stage 2	Onboarding Jour...
<input type="checkbox"/> Patrick Kitchell	Onboarding J...	9a81f01e-5f48-f0...	Onboarding Stage 1	Onboarding Jour...

### BPF History Status Reason:

There have been configured status reasons for logging BPFs and they are as follows:

**Start:** When the BPF Instance on the Primary Table was Created

**Active:** Current Active Table

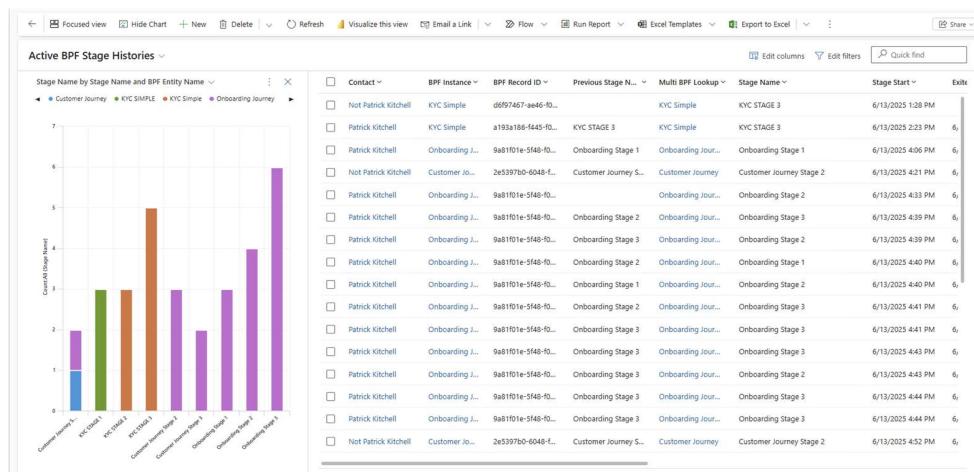
**In the Past:** Stages in the BPF already completed

**Finished:** When the BPF is completed

<input type="checkbox"/> Contact ▾	BPF Instance ▾	Status ▾	Status Reason ▾
<input type="checkbox"/> Not Patrick Kitchell	KYC Simple	Active	Active
<input type="checkbox"/> Patrick Kitchell	KYC Simple	Active	BPF Finished
<input type="checkbox"/> Patrick Kitchell	Onboarding Journey	Active	In The Past
<input type="checkbox"/> Not Patrick Kitchell	Customer Journey	Active	In The Past
<input type="checkbox"/> Patrick Kitchell	Onboarding Journey	Active	In The Past
<input type="checkbox"/> Patrick Kitchell	Onboarding Journey	Active	In The Past
<input type="checkbox"/> Patrick Kitchell	Onboarding Journey	Active	In The Past
<input type="checkbox"/> Patrick Kitchell	Onboarding Journey	Active	In The Past
<input type="checkbox"/> Patrick Kitchell	Onboarding Journey	Active	In The Past

## Added Some Simple Graphs

Once this table is created and there is more detail on what is in the table then I can also create some graphs against the attributes. The one in the picture below shows **Status and BPF Type**



## Duration

One sticking point is tracking time and in the current data model the time on stages is overwritten and with the BPF Historical Table - I can track the duration of the BPF between stages, which gets lost in the current data model and complete time

Last Stage Duration	Duration (BPF Reference)
	2.05 days
27.00	
27.00	

## Contacts

On the view for Contact - I added the editable grid component and showed the BPF historical stages for each contact and I have added to the Contact itself in a new tab as a sub grid

## Contact View

Active Contacts							grid
Group By:	(no grouping)	First Name	Last Name	Email	Company Name	Business Phone	GDPR Optout
✓	Full Name ↑	A Goofy	Mule	---	---	---	No
>	A Goofy Mule	A Goofy	Mule	---	---	---	No
>	Donald Duck	Donald	Duck	eefe@dfdf.j	---	---	No
>	Micky Mouse	Micky	Mouse	mm@mm.d	---	---	No
>	Not Patrick Kitchell	Not Patrick	Kitchell	---	---	---	No
>	Patrick Kitchell	Patrick	Kitchell	patrick.kitchell@gmail.dk	Kitchell A/S	---	No

## On the Contact Itself

The screenshot shows a contact record for 'A Goofy Mule'. The 'Business Processes' tab is active. A red box highlights the 'Status Reason' column in the BPF Historical Table, which lists three entries: 'Start', 'Start', and 'Start'.

Created On	Name	BPF Instance	Stage Name	Stage Start	Status Reason
6/15/2025 9:23 PM	Customer Journey	Customer Journey	Customer Journey Stage 1	6/15/2025 9:23 PM	Start
6/15/2025 9:23 PM	Onboarding Journey	Onboarding Journ...	Onboarding Stage 1	6/15/2025 9:22 PM	Start
6/15/2025 9:17 PM	KYC SIMPLE	KYC Simple	KYC STAGE 1	6/15/2025 9:17 PM	Start

## Deep Dive Part 2 Conclusions

Well, there is a bit more to do with BPFs because I want to drive value from a better understanding of how everything is tied together. This is important to help find any missing pieces needed to drive end user business value as well as possibly re-imagine old processes into digital ones.

### In my next post I will show the following:

- BPF Historical Table in Detail
- Power Automate Flows to drive this
- Reporting
- Possible Canvas App to visualize all this

## Tools

### XRM Toolbox

<https://www.xrmtoolbox.com/>

### ERD Maker

<https://github.com/YonatanCohavi/ERDMaker>

## User Case, Build and Other Findings

My goal has been with these 3 posts to create a better understanding of the mechanisms of Business Process Flows (BPF)

This post is to round off what I have done and explain in more detail the ideas and inspirations with BPFs and how my own curiosity can see good possibilities for value from using multiple BPFs to help drive and visualize processes.

### In my solution I have created the following:

1. **A Process App** to see the BPF History Table and other Tables in the BPF Data Mode
2. **A BPF History Table** to Track Changes in BPFs from Start to Finish and Stage Movements
3. **A Canvas App** to try and visualize the BPF History Table - I admit it is not "nice" but food for future exploration
4. **In - App Notification** on the completion of a BPF Flow
5. **Power Automate Flows** manage the creation of the BPF Stage History Lines
6. **Rudimentary Power BI report** - Also food for future exploration

### 1.) Process APP

I created this Process App to Monitor what was possible as mentioned in the prior post.

Created On	Contact	WorkID	BPF Instance	BPF Record ID	Stage Name
6/13/2025 1:28 PM	Not Patrick Kitchell	13b7c019-ef45-f0...	KYC Simple	d6f97467-ae46-f0...	KYC STAGE 3
6/13/2025 4:21 PM	Not Patrick Kitchell	07c9e44c-5c48-f0...	Customer Jo...	2e5397b0-6048-f...	Customer Journey Stage 2
6/13/2025 4:52 PM	Not Patrick Kitchell	07c9e44c-5c48-f0...	Customer Jo...	2e5397b0-6048-f...	Customer Journey Stage 2
6/13/2025 4:52 PM	Not Patrick Kitchell	07c9e44c-5c48-f0...	Customer Jo...	2e5397b0-6048-f...	Customer Journey Stage 1
6/13/2025 4:53 PM	Not Patrick Kitchell	07c9e44c-5c48-f0...	Customer Jo...	2e5397b0-6048-f...	Customer Journey Stage 2
6/13/2025 4:53 PM	Not Patrick Kitchell	07c9e44c-5c48-f0...	Customer Jo...	2e5397b0-6048-f...	Customer Journey Stage 3
6/13/2025 4:53 PM	Not Patrick Kitchell	07c9e44c-5c48-f0...	Customer Jo...	2e5397b0-6048-f...	Customer Journey Stage 3
6/15/2025 9:20 AM	Micky Mouse	13b7c019-ef45-f0...	KYC Simple	b0756e25-2749-f...	KYC STAGE 2
6/15/2025 9:31 AM	Donald Duck	13b7c019-ef45-f0...	KYC Simple	cec2311e-ba49-f0...	KYC STAGE 1
6/15/2025 9:32 AM	Donald Duck	13b7c019-ef45-f0...	KYC Simple	cec2311e-ba49-f0...	KYC STAGE 2
6/15/2025 9:33 AM	Micky Mouse	13b7c019-ef45-f0...	KYC Simple	b0756e25-2749-f...	KYC STAGE 3
6/15/2025 9:34 AM	Donald Duck	13b7c019-ef45-f0...	KYC Simple	cec2311e-ba49-f0...	KYC STAGE 3

## 2.) BPF History Table

This is the view over the BPF History Table in my Process App. These are created when a

- BPF is created
- The stage is changed
- The BPF is Finished

All the information in this view and the forms below come from the data model and grabbed via power automate

Active BPF Stage Histories ▾								
	Created On ↑	Contact	WorkID	BPF Instance	BPF Record ID	Stage Name	Stage ID	Stage Sta
<input type="checkbox"/>	6/13/2025 1:28 PM	Not Patrick Kitchell	13b7c019-ef45-f0...	KYC Simple	d6f97467-ae46-f0...	KYC STAGE 3	89a593a9...	6/13/202!
<input type="checkbox"/>	6/13/2025 4:21 PM	Not Patrick Kitchell	07c9e44c-5c48-f0...	Customer Jo...	2e5397b0-6048-f...	Customer Journey Stage 2	2770a37b...	6/13/202!
<input type="checkbox"/>	6/13/2025 4:52 PM	Not Patrick Kitchell	07c9e44c-5c48-f0...	Customer Jo...	2e5397b0-6048-f...	Customer Journey Stage 2	2770a37b...	6/13/202!
<input type="checkbox"/>	6/13/2025 4:52 PM	Not Patrick Kitchell	07c9e44c-5c48-f0...	Customer Jo...	2e5397b0-6048-f...	Customer Journey Stage 1	01f1dce7...	6/13/202!
<input type="checkbox"/>	6/13/2025 4:53 PM	Not Patrick Kitchell	07c9e44c-5c48-f0...	Customer Jo...	2e5397b0-6048-f...	Customer Journey Stage 2	2770a37b...	6/13/202!
<input type="checkbox"/>	6/13/2025 4:53 PM	Not Patrick Kitchell	07c9e44c-5c48-f0...	Customer Jo...	2e5397b0-6048-f...	Customer Journey Stage 3	2c199637...	6/13/202!
<input type="checkbox"/>	6/13/2025 4:53 PM	Not Patrick Kitchell	07c9e44c-5c48-f0...	Customer Jo...	2e5397b0-6048-f...	Customer Journey Stage 3	2c199637...	6/13/202!
<input type="checkbox"/>	6/15/2025 9:20 AM	Micky Mouse	13b7c019-ef45-f0...	KYC Simple	b0756e25-2749-f...	KYC STAGE 2	ee7e2acd...	6/15/202!
<input type="checkbox"/>	6/15/2025 9:31 AM	Donald Duck	13b7c019-ef45-f0...	KYC Simple	cec2311e-ba49-f0...	KYC STAGE 1	e716e4e9...	6/15/202!
<input type="checkbox"/>	6/15/2025 9:32 AM	Donald Duck	13b7c019-ef45-f0...	KYC Simple	cec2311e-ba49-f0...	KYC STAGE 2	ee7e2acd...	6/15/202!
<input type="checkbox"/>	6/15/2025 9:33 AM	Micky Mouse	13b7c019-ef45-f0...	KYC Simple	b0756e25-2749-f...	KYC STAGE 3	89a593a9...	6/15/202!
<input type="checkbox"/>	6/15/2025 9:34 AM	Donald Duck	13b7c019-ef45-f0...	KYC Simple	cec2311e-ba49-f0...	KYC STAGE 3	89a593a9...	6/15/202!
<input type="checkbox"/>	6/15/2025 5:49 PM	Donald Duck	13b7c019-ef45-f0...	KYC Simple	cec2311e-ba49-f0...	KYC STAGE 2	ee7e2acd...	6/15/202!
<input type="checkbox"/>	6/15/2025 9:02 PM	Donald Duck	13b7c019-ef45-f0...	KYC Simple	cec2311e-ba49-f0...	KYC STAGE 3	89a593a9...	6/15/202!
<input type="checkbox"/>	6/15/2025 9:17 PM	A Goofy Mule	13b7c019-ef45-f0...	KYC Simple	0d247b6c-1d4a-f...	KYC STAGE 1	e716e4e9...	6/15/202!
<input type="checkbox"/>	6/15/2025 9:23 PM	A Goofy Mule	7aca0a3b-2d46-f...	Onboarding J...	61857f72-1d4a-f0...	Onboarding Stage 1	b9254829...	6/15/202!

### Header:

In the Header I have added the Owner and I have added Status Reasons that were mentioned earlier. Just a refresher - On the Status Reason Field itself, I have added the following reasons to reflect the BPF History. This will be used later in my rudiment Power BI and Canvas APP

The screenshot shows a form interface for managing Business Process Flows (BPF). At the top, it displays the owner as 'Ameed Sheikh' and the status as 'Active'. Below this, there is a section for 'Status Reason' which is currently set to 'In The Past'. A dropdown menu is open, listing several options: 'Start', 'Active', 'In The Past' (which is checked and highlighted with a red box), and 'BPF Finished'. Other fields visible include 'Owner' and 'Status'.

### Top Form:

I have added the following fields to the BPF History Form starting with the top of the form.

The screenshot displays the top portion of the BPF History Form. It includes the following fields and their values:

- BPF Information**
- BPF Name:** Customer Journey
- Contact:** Not Patrick Kitchell
- BPF Instance ID:** 2e5397b0-6048-f011-8779-7ced8d4960a8 (highlighted with a red box)
- BPF Instance Lookup:** Customer Journey
- Multi BPF Lookup:** Customer Journey
- WorkflowID:** 07c9e44c-5c48-f011-877a-7c1e52762e83 (highlighted with a red box)

The BPF History Form Top contains

- BPF Name
- A Lookup to Contacts (Primary Entity)
- BPF Instance ID
- A Lookup to the BPF Instance

- A Polymorphic Lookup (Mult BPF Lookup) to various BPFs
- The Workflow ID

## Bottom Form

To the bottom of the form I have the following Information

BPF Stages		
Stage Name	Customer Journey Stage 2	
Stage ID	2770a37b-3393-48dd-8f15-e1af6f01367b	
Previous Stage Name	Customer Journey Stage 1	
Previous Stage ID	01f1dce7-72c5-4761-b073-b7d090de9c7f	
Stage Start	6/13/2025	4:21 PM
Exited On	6/13/2025	4:48 PM
Last Stage Duration	27.00	

## BPF History Form Bottom

- Stage Name
- Stage ID
- Previous Stage Name
- Previous Stage ID
- Stage Start Date /Time
- Stage Exit Date/Time
- Duration Between Stages(Calculated Field)

## Look up Contact

Because of the look up on contacts I can add at a sub view to each contact form to show the various BPFs and their Status Reasons.

The screenshot shows a Business Process Flow (BPF) history grid. At the top, there's a header bar with tabs: Summary, Details, Originating Lead, Business Processes (which is selected), Visual BPF, and Related. Below the header is a table with the following data:

<input type="checkbox"/>	Created On ↓	Name	BPF Instance	Stage Name	Stage Start	Previo
<input type="checkbox"/>	6/13/2025 4:53 PM	Onboarding Journey	Customer Journey	Customer Journey Stage 3	6/13/2025 4:53 PM	Custor
<input type="checkbox"/>	6/13/2025 4:53 PM	Onboarding Journey	Customer Journey	Customer Journey Stage 3	6/13/2025 4:53 PM	Custor
<input type="checkbox"/>	6/13/2025 4:53 PM	Onboarding Journey	Customer Journey	Customer Journey Stage 2	6/13/2025 4:53 PM	Custor
<input type="checkbox"/>	6/13/2025 4:52 PM	Onboarding Journey	Customer Journey	Customer Journey Stage 1	6/13/2025 4:52 PM	Custor
<input type="checkbox"/>	6/13/2025 4:52 PM	Onboarding Journey	Customer Journey	Customer Journey Stage 2	6/13/2025 4:52 PM	Custor
<input type="checkbox"/>	6/13/2025 4:21 PM	Onboarding Journey	Customer Journey	Customer Journey Stage 2	6/13/2025 4:21 PM	Custor
<input type="checkbox"/>	6/13/2025 1:28 PM	KYC Simple	KYC Simple	KYC STAGE 3	6/13/2025 1:28 PM	

Rows: 7

## Contact BPF History

### Active Views with Embedded Sub grids

I am not that wild about this because it is an editable grid where these BPF History Lines will be read only. It is just show what is possible.

The screenshot shows an Active Contacts view. At the top, there's a header bar with tabs: Active Contacts (selected), Details, Originating Lead, Business Processes, Visual BPF, and Related. Below the header is a table with the following data:

Active Contacts				
Group By: (no grouping)				
✓   Full Name ↑	First Name	Last Name	Email	
✓ A Goofy Mule	A Goofy	Mule	eferfe@dfd.dl	
<input type="checkbox"/>   Created On ↓	Name	BPF Instance	Stage Name	Stage Start
6/23/2025 8:41 AM	KYC Simple	KYC Simple	KYC STAGE 3	6/23/2025 8:41 AM
6/23/2025 8:41 AM	KYC Simple	KYC Simple	KYC STAGE 2	6/23/2025 8:41 AM
6/20/2025 2:11 PM	Onboarding Journey	Customer Journey	Customer Journey Stage 2	6/20/2025 2:11 PM
6/15/2025 9:23 PM	Customer Journey	Customer Journey	Customer Journey Stage 1	6/15/2025 9:23 PM
6/15/2025 9:23 PM	Onboarding Journey	Onboarding Journey	Onboarding Stage 1	6/15/2025 9:22 PM
6/15/2025 9:17 PM	KYC SIMPLE	KYC Simple	KYC STAGE 1	6/15/2025 9:17 PM

## 3.) Canvas App

This is still in the dream stage, I would try to build my first PCF control on this instead of using canvas up.

I want to see how much I can grab in a gallery on a Canvas App. I started with embedding an app via the canvas app component in Model Driven apps.

This gives us the ability to use ModelDrivenFormIntegration, which makes life easier.



## ModelDrivenFormIntegration

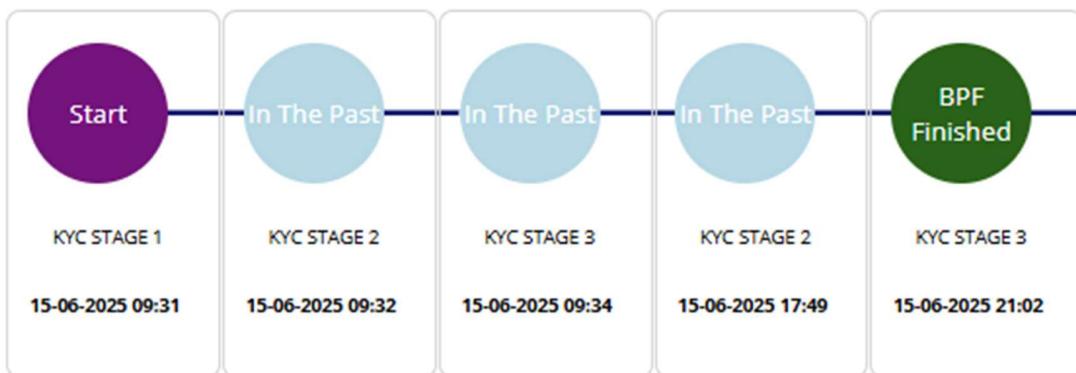
### Gallery 1:

Just a simple gallery with contact + BPF history information

<b>Donald Duck</b>	BPF Record ID:	cec2311e-ba49-f011-877a-7c1e52344f57	15-06-2025 09:31
KYC SIMPLE(ID)	13b7c019-ef45-f011-8779-7c1e5236ac0a		
KYC STAGE 1	e716e4e9-c2eb-475d-a379-31c63efedc80	15-06-2025 09:31	
		15-06-2025 09:32	
<b>Active</b>		<b>Start</b>	
<b>Donald Duck</b>	BPF Record ID:	cec2311e-ba49-f011-877a-7c1e52344f57	15-06-2025 09:32
KYC Simple(ID)	13b7c019-ef45-f011-8779-7c1e5236ac0a		
KYC STAGE 2	ee7e2acd-09d6-45e3-8a9c-a21320774b69	15-06-2025 09:32	
KYC STAGE 1	e716e4e9-c2eb-475d-a379-31c63efedc80	15-06-2025 09:30	
<b>Active</b>		<b>In The Past</b>	
<b>Donald Duck</b>	BPF Record ID:	cec2311e-ba49-f011-877a-7c1e52344f57	15-06-2025 09:34
KYC Simple(ID)	13b7c019-ef45-f011-8779-7c1e5236ac0a		
KYC STAGE 3	89a593a9-52fc-4d61-80fa-0dd43dc63a08	15-06-2025 09:34	
KYC STAGE 2	ee7e2acd-09d6-45e3-8a9c-a21320774b69	15-06-2025 17:49	
<b>Active</b>		<b>In The Past</b>	

## Gallery 2:

**Same Gallery just visualized:**



## Gallery 2 Embedded

I embedded in a tab on contact. I am not satisfied with this however it gives food for thought and ideas for future enhancements. I think about [Allan De Castro](#) and his old PCF control that I love - [PCF Gallery](#) or [GitHub](#) - the PCF control is out of date and gives an indication of what can be done.

The screenshot shows a Salesforce contact record for "A Goofy Mule". The top navigation bar includes "Summary", "Details", "Originating Lead", "Business Processes", "Visual BPF" (which is underlined), and "Related".

**KYC Simple** (Completed in 7 days) is displayed above the first timeline.

**Visual BPF** section:

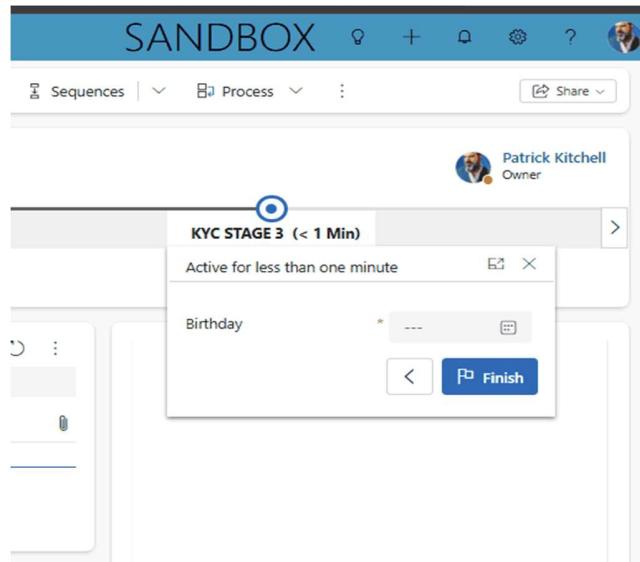
- KYC STAGE 1:** Status: In Progress (blue checkmark), Last Update: 15-06-2025 21:17, KYC STAGE 1.
- KYC STAGE 2:** Status: In Progress (blue checkmark), Last Update: 23-06-2025 08:41, KYC STAGE 2.
- KYC STAGE 3:** Status: Completed (green checkmark), Last Update: 23-06-2025 08:41, KYC STAGE 3.

**Customer Journey** section:

- Onboarding Stage 1:** Status: Start, Last Update: 15-06-2025 21:23.
- Customer Journey Stage 1:** Status: Active, Last Update: 20-06-2025 14:11.

## 4.) In App Notification.

I also created an In App Notification at the 1c. KYC SIMPLE Finished (see Power Automate Below)



## 5.) Power Automate

I have worked towards figuring out how to create, update and finish the BPF Histories and there needs to be more work to optimize these flows and maybe into 1 flow. I didn't get that far to have a flow that can grab any BPF. I have created 3 flows for each BPF in my scenario. 3 Flows to Each KYC

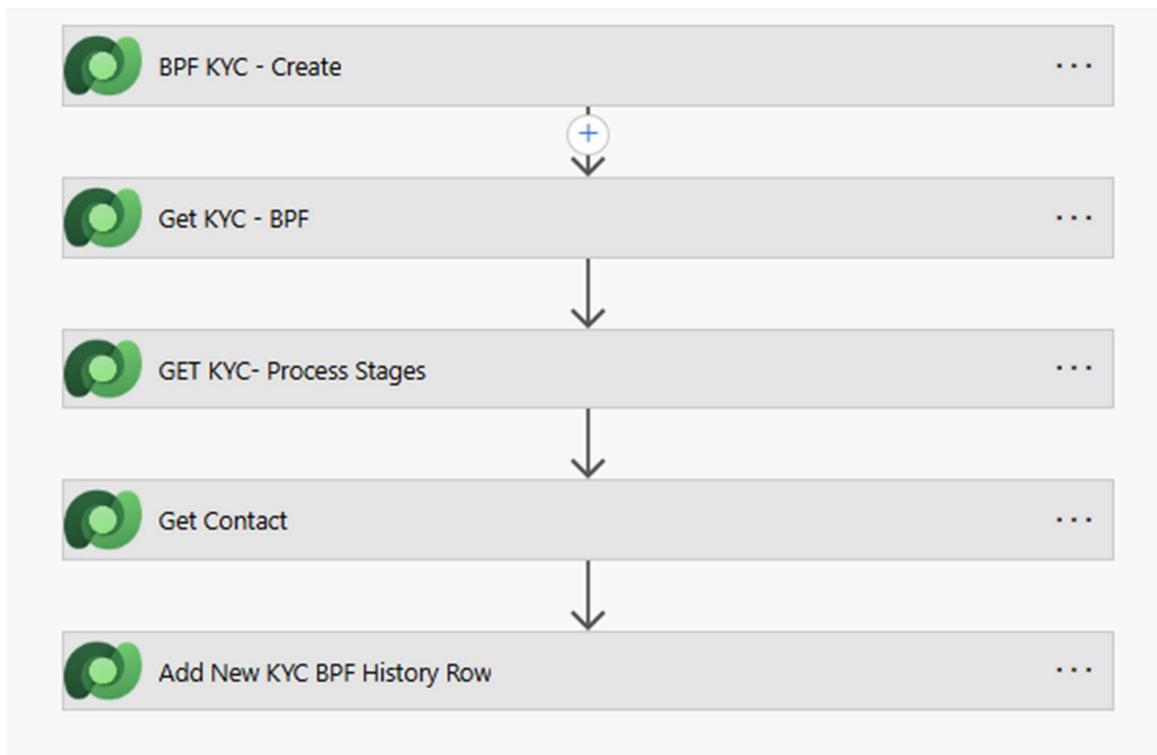
## BPF Approvals &gt; Cloud flows

Display name ↑ ↓

- <sup>a</sup> 1a. Create KYC
- <sup>a</sup> 1b. Log KYC SIMPLE Stage Transition
- <sup>a</sup> 1c. KYC SIMPLE Finished
- <sup>a</sup> 2a. Create Customer Journey
- <sup>a</sup> 2b. Log Customer Journey Stage Transition
- <sup>a</sup> 2c .Customer Journey 1 Finished
- <sup>a</sup> 3a. Create Onboarding
- <sup>a</sup> 3b. Log Onboarding Stage Transition
- <sup>a</sup> 3c. Onboarding Finished

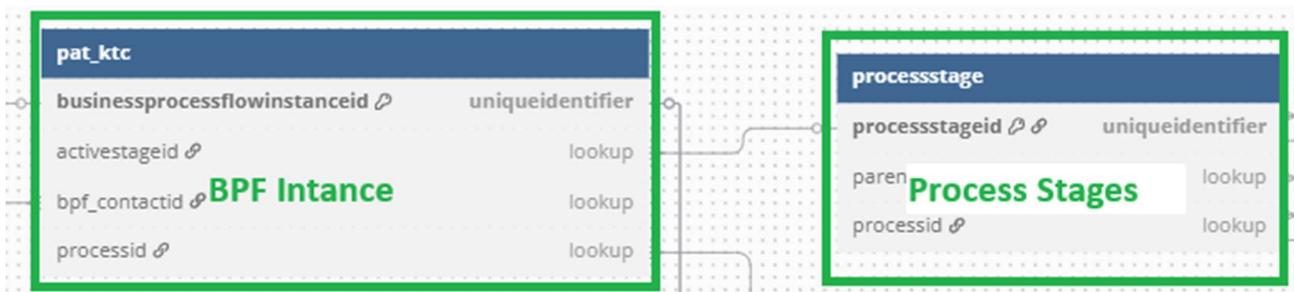
## 1a. Create KYC - Start Power Automate:

This creates a BPF History line with the **status reason = "Start"**



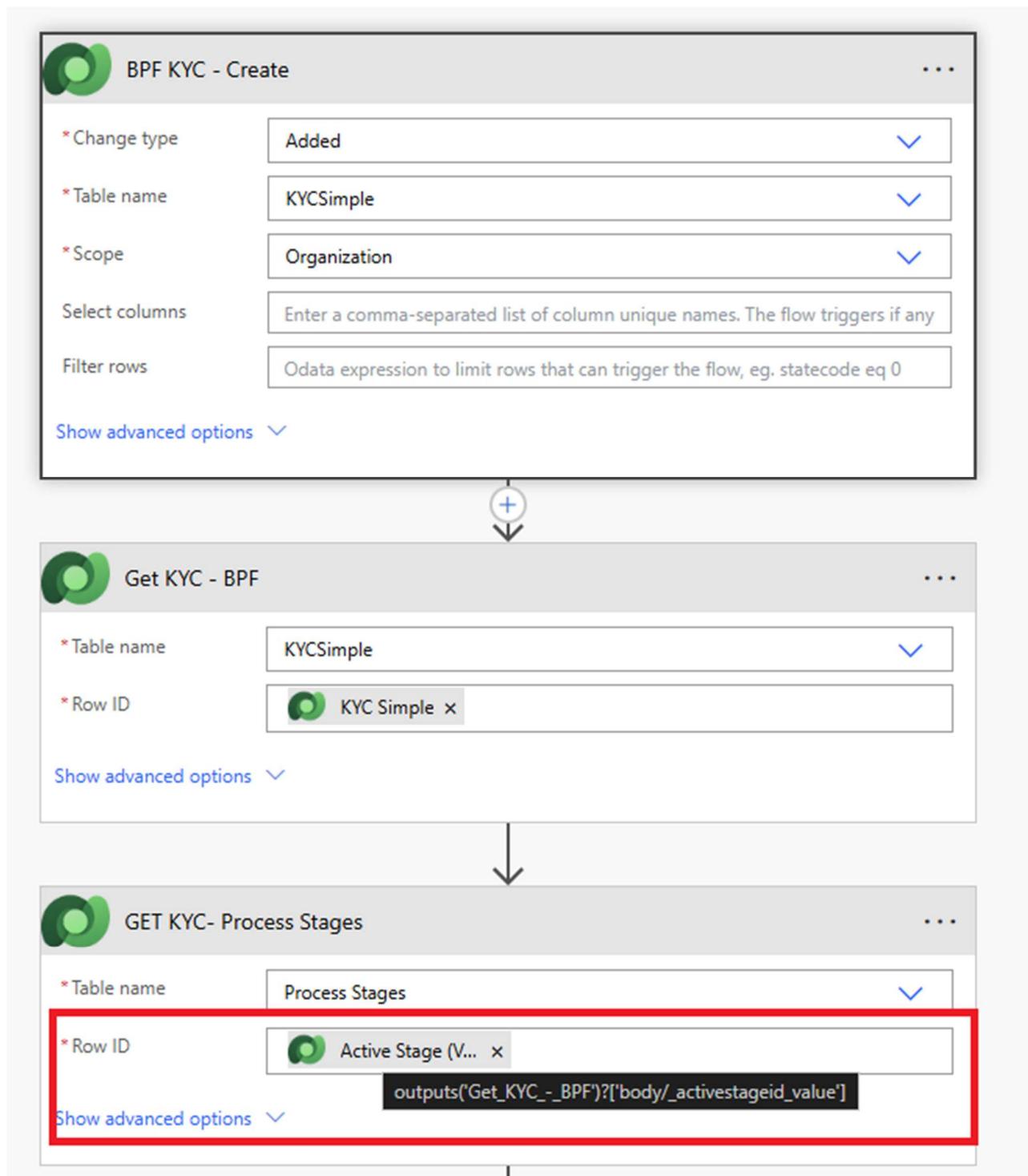
## 1a. Trigger Flow when an Instance of the KYC Simple is Created

What is important is to get the active process stages. The BPF is one table and the Process Stage is another Table.



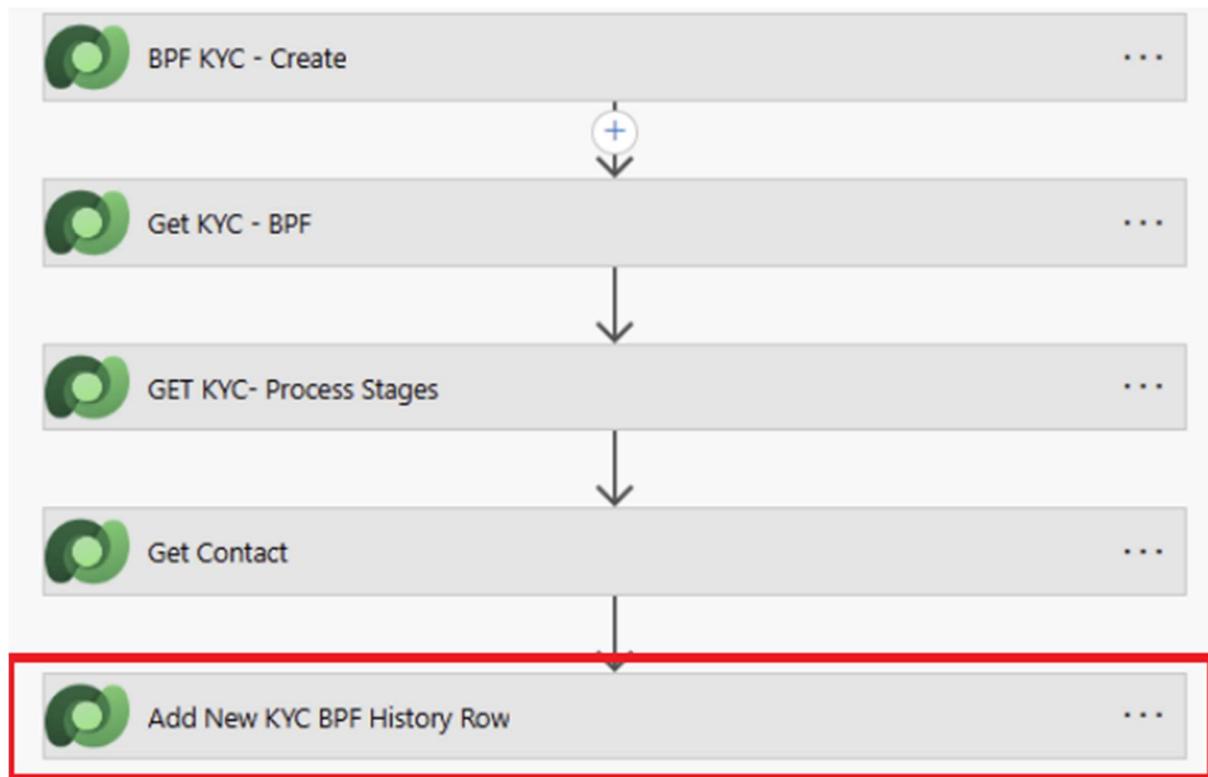
## 1a. Create KYC

Triggers the flow and grabs the BPF, Stages and Contact Information



## Getting Process ID

Adds a new BPF History with information from BPF Instance, Stages and Contact and Adding a new BPF History Row



## Process ID for the BPF Instance

Add New KYC BPF History Row

\*Table name: BPF Stage Histories

Return Full Metadata: The header parameter to customer to opt-in returning full odata metadata

BPF Entity Name: KYC SIMPLE

BPF Instance (Processes): workflows(KYC Simple)

BPF Record ID: outputs('Get\_KYC\_-\_BPF')?['body/\_processid\_value']

BPF Reference (KYCSimple):

BPF Stage History: Unique identifier for entity instances

BPF Template Name: Process (Tabeln...)

Business Process (Business Processes):

Contact (Contacts): contacts(Contact)

## Create a Start BPF Line

Adding Stage Names, Times, and Status Reason and Create Start BPF Line with Information

Name	KYC SIMPLE
Owner (Owners)	Owner Id
Previous Stage ID	Previous Stage ID
Previous Stage Name	Previous Stage Name
Record Created On	Date and time that the record was migrated.
Regarding (Accounts)	
Regarding (Appointments)	
Regarding (Cases)	
Regarding (Contacts)	
Regarding (Leads)	
Regarding (Opportunities)	
Regarding (Phone Calls)	
Regarding (Quotes)	
Regarding (Tasks)	
Stage ID	 Active Stage (V... ×)
Stage Name	 Process Stage ... ×
Stage Start	 utcNow() ×
Status Reason	Start 
Time Zone Rule Version Number	For internal use only.
User (Users)	
UTC Conversion Time Zone Code	Time zone code that was in use when the record was created.
WorkID	 Process (Værdi) ×

## 1b. Log KYC SIMPLE Stage Transition

This is the flow I created that triggers on the **modification of the BPF KYC Simple**.



After the flow grabs the BPF instance than it filters for the last BPF History using filtered rows

The screenshot shows the configuration for a 'List the Last PBF History Row' step. The step has the following settings:

- Table name:** BPF Stage Histories
- Select columns:** Enter a comma-separated list of column unique names to limit which columns are returned.
- Filter rows:** pat\_bpfreccordid eq KYC Simple
- Sort By:** pat\_stagestarttest desc
- Fetch Xml Query:** Enter a Fetch XML query for advanced customization.
- Row count:** 1
- Show advanced options:** A dropdown menu.

Filtering with the following:

```
pat_bpfreccordid eq @{outputs('Get_KYC_Simple_-_
_BPF')?['body/businessprocessflowinstanceid']}
```

*pat\_stagestarttest desc*

The screenshot shows the configuration for a variable named 'var previous stageid'. The variable has the following settings:

- Name:** var\_previous\_stageid
- Type:** String
- Value:** first(outputs('List\_the\_Last\_PBF\_History\_Row')?['body'][value])?['pat\_stageid']

Variable for StageID

```
first(outputs('List_the_Last_PBF_History_Row')?['body'][value])?['pat_stageid']
```

{x} var previous stagename

\*Name: var\_previous\_stagename

\*Type: String

Value: `first(outputs('List_the_Last_PBF_History_Row')?['body']['value'])?['pat_stagename']`

## Variable for StageName

`first(outputs('List_the_Last_PBF_History_Row')?['body']['value'])?['pat_stagename']`

Is There a Prior BPF History Row

`length(outputs('List_the_Last_PBF_History_Row')?['body'])?['value'])`

**This Variable returns a number — the number of rows retrieved by the List\_the\_Last\_PBF\_History\_Row action.**

`length(outputs('List_the_Last_PBF_History_Row')?['body'])?['value'])`

## If Greater than "0" - Update Prior Row

Is There a Prior BPF History Row

`length(outputs('List_the_Last_PBF_History_Row')?['body'])?['value'])`

If yes:

- Update Exit Time and Status Reason set to In The Past
- Select an output from previous steps: value
- Update BPF History Row

If no:

Add an action

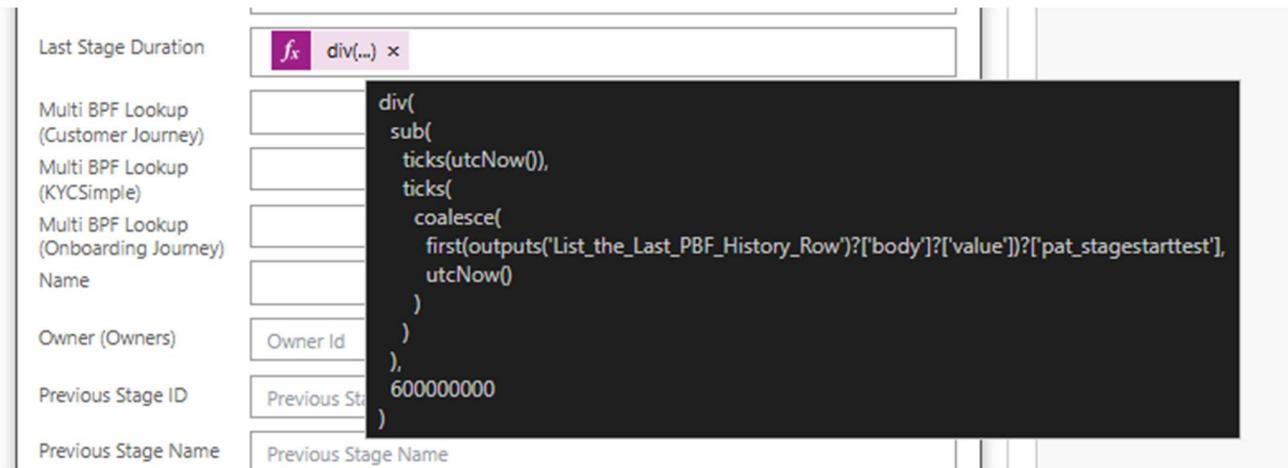
## Conditions

The screenshot shows a Power Automate 'If yes' condition step. It contains two actions:

- Action 1: Update Exit Time and Status Reason set to In The Past**
  - \*Select an output from previous steps
  - value
- Action 2: Update BPF History Row**
  - \*Table name: BPF Stage Histories
  - \*Row ID: BPF Stage Hist...
  - Return Full Metadata: The header parameter to customer to opt-in returning full odata metadata
  - BPF Entity Name: Logical name of BPF entity table
  - BPF Instance (Processes):
  - BPF Record ID: GUID of BPF instance (any process)
  - BPF Reference (KYCSimple):
  - BPF Template Name: Friendly BPF template name
  - Business Process (Business Processes):
  - Contact (Contacts):
  - Duration Fixed:
  - Exited On: `utcNow()`
  - Last Stage Duration: `div(...)`

## Update Prior Row

Last Stage Duration is also a Variable that the **number of minutes** since pat\_stagestarttest. If the value is missing, it returns **0**.



```

Last Stage Duration


fx div(...)

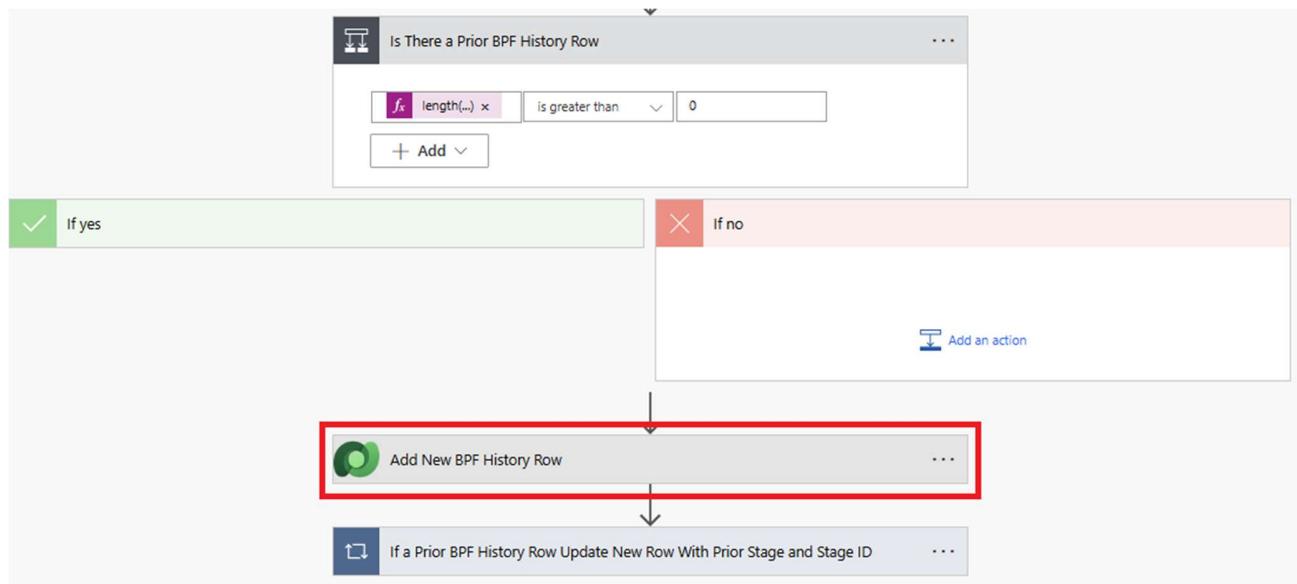
    div(
        sub(
            ticks(utcNow()),
            ticks(
                coalesce(
                    first(outputs('List_the_Last_PBF_History_Row')?['body']?['value'])?['pat_stagetest'],
                    utcNow()
                )
            ),
            600000000
        )
    )


```

Multi BPF Lookup (Customer Journey)  
 Multi BPF Lookup (KYCSimple)  
 Multi BPF Lookup (Onboarding Journey)  
 Name  
 Owner (Owners) Owner Id  
 Previous Stage ID Previous Stage ID  
 Previous Stage Name Previous Stage Name

## Lastly Set Status Reason to "In The Past"

After Condition - Add New History Row - Sames as above



## Lastly, If a Prior BPF History Then Update Previous Stage ID and Stage Name with Variables

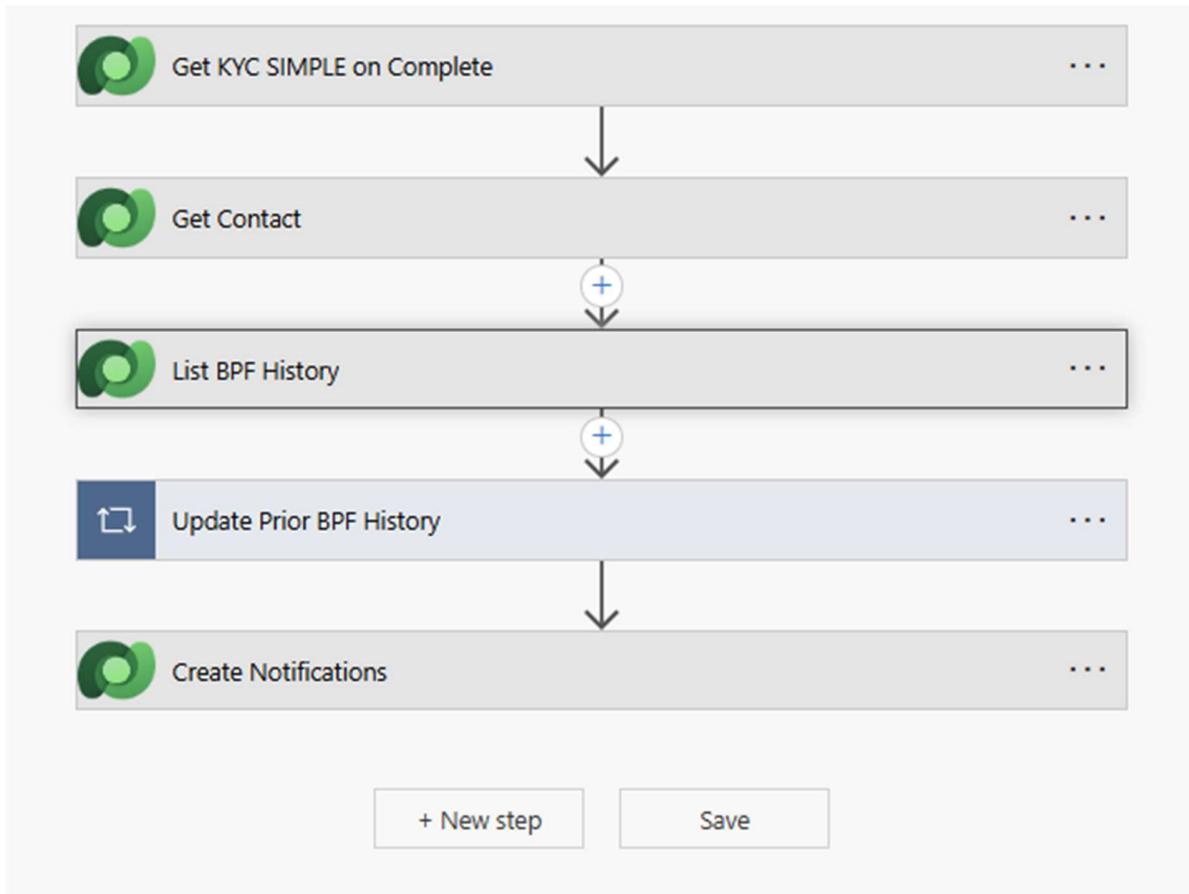
Update Previous Stage ID + Name with Variables.

The screenshot shows the configuration of a Power Automate step titled "If a Prior BPF History Row Update New Row With Prior Stage and Stage ID". The step has a single input "value" selected from previous steps. The main configuration pane is for the "Update New BPF History Row With Prior Stage and Stage ID" action. It includes fields for:

- \*Table name: BPF Stage Histories
- \*Row ID: BPF Stage Hist...
- Return Full Metadata: The header parameter to customer to opt-in returning full odata metadata
- BPF Entity Name: Logical name of BPF entity table
- BPF Instance (Processes):
- BPF Record ID: GUID of BPF instance (any process)
- BPF Reference (KYCSimple):
- BPF Template Name: Friendly BPF template name
- Business Process (Business Processes):
- Contact (Contacts):
- Duration Fixed:
- Exited On:
- Last Stage Duration:
- Multi BPF Lookup (Customer Journey):
- Multi BPF Lookup (KYCSimple):
- Multi BPF Lookup (Onboarding Journey):
- Name:
- Owner (Owners): Owner Id
- Previous Stage ID: {x} var\_previous\_st... x
- Previous Stage Name: {x} var\_previous\_st... x

## 1c. KYC SIMPLE Finished

Triggered when the BPF Instance is completed using a filtered lookup



**Triggers on modified and gets the row with statuscode =2**

* Change type	Modified
* Table name	KYCSimple
* Scope	Organization
Select columns	statuscode
Filter rows	statuscode eq 2
<a href="#">Show advanced options</a>	

*Using Statuscode*

*"statuscode eq 2"*

## Filtering away the BPF History to the last row

Filter rows = Statuscode eq 2

The screenshot shows the 'List BPF History' dialog box. It has the following settings:

- Table name: BPF Stage Histories
- Select columns: (empty)
- Filter rows: pat\_bpfreccordid eq 'KYC Simple'
- Sort By: pat\_stagename asc
- Fetch Xml Query: (empty)
- Row count: 1

At the bottom left, there is a link 'Show advanced options'.

Updates the last BPF History:

Update Prior BPF

- Exit time = **UTCNow()**,
- **Previous Stage ID and Stage Name and**
- New Status Reason = **BPF Finished**

Update Prior BPF History

\*Select an output from previous steps  
 value x

**BPF Stage Histories**

*Table name	BPF Stage Histories
*Row ID	 BPF Stage Hist... <span style="float: right;">x</span>
Return Full Metadata	The header parameter to customer to opt-in returning full odata metadata
BPF Entity Name	Logical name of BPF entity table
BPF Instance (Processes)	
BPF Record ID	GUID of BPF instance (any process)
BPF Reference (KYCSimple)	
BPF Template Name	Friendly BPF template name
Business Process (Business Processes)	
Contact (Contacts)	
Duration Fixed	
Exited On	 utcNow() <span style="float: right;">x</span>
Last Stage Duration	
Multi BPF Lookup (Customer Journey)	
Multi BPF Lookup (KYCSimple)	
Multi BPF Lookup (Onboarding Journey)	
Name	
Owner (Owners)	Owner Id
Previous Stage ID	 Stage ID <span style="float: right;">x</span>
Previous Stage Name	 Stage Name <span style="float: right;">x</span>
Progression (Associations)	

Finally, Creates in app notification that fire. See #4 above.

Field	Description
*Table name	Notifications
Return Full Metadata	The header parameter to customer to opt-in returning full odata metadata
Title	Full Name x Has Completed KYC
Body	Full Name x Has Completed KYC Completed On x
Data	Custom data for the notification that can be used by the notification card
Expiry (seconds)	60
IconType	Info
Import Sequence Number	Sequence number of the import that created this record.
Model-driven app (Model-driven Apps)	This field is not used
Notification	Unique identifier for entity instances
Owner (Owners)	Owner Id
Partition Id	Partitioning will be based on owner and it is recommended to specify this field
Priority	High
Record Created On	Date and time that the record was migrated.
Time Zone Rule Version Number	For internal use only.
Toast Type	Timed
UTC Conversion Time Zone Code	Time zone code that was in use when the record was created.
<a href="#">Hide advanced options</a>	

## 6.) Power BI

This is where I stopped and haven't spent any time more looking into this and might in the future make some cool reporting on the tool of the BPF History table and other tables that can be used..

... - - - - -

---

BPF Instance × Stage Start × Exited On ×

Count of Contact  
2

## Conclusion

Like most Dataverse, Power Apps there are many possibilities with everything. I have always been intrigued by BPFs and can see a wide range of value driven possibilities for BPFS. I hope this little deep dive made someone curious like am curious towards this technology.