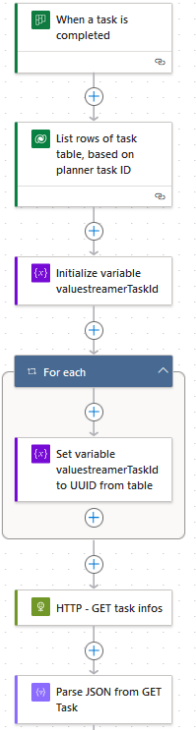


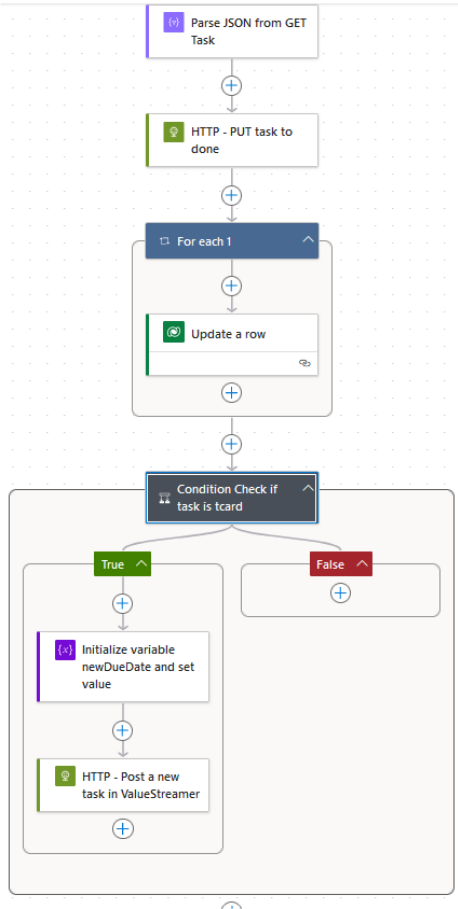
Technical process (detailed) of flow "Update ValueStreamer task when Planner task is completed"

This page shows the process of the flow with all details.

Process Overview

Step	Screenshot
1-7	 The screenshot displays a Power Automate flow diagram. It begins with a trigger 'When a task is completed'. This is followed by an action 'List rows of task table, based on planner task ID'. Next is 'Initialize variable valuestreamerTaskId'. A 'For each' loop follows, containing the action 'Set variable valuestreamerTaskId to UUID from table'. After the loop, there is an action 'HTTP - GET task infos', and finally, 'Parse JSON from GET Task'. Each step is connected to the next by a downward arrow with a plus sign in a circle.

7-11.1.2



Step by step

Step	Description	Power Automate Function: Connector, Display-Name	Variables, Parameters, Formula						
1.	When a task is completed in planner, then trigger this flow.	Trigger: Planner, When a task is completed	Parameter <table><tr><th>Key</th><th>Value</th></tr><tr><td>Group Id:</td><td>YourGroupId</td></tr><tr><td>Plan Id:</td><td>YourPlanId</td></tr></table>	Key	Value	Group Id:	YourGroupId	Plan Id:	YourPlanId
Key	Value								
Group Id:	YourGroupId								
Plan Id:	YourPlanId								
2.	Get the completed task in planner from the task table.	List rows: Microsoft Dataverse, List rows of task table, based on planner task ID	Parameter <table><tr><th>Key</th><th>Value</th></tr></table>	Key	Value				
Key	Value								

			Table Name: YourTaskTable Filter Rows: cr4b3_planner eq triggerOutputs()?['body/id']												
3.	Create a new variable called valuestreamerTaskId of type String and with no value.	Initialize Variable: Variables, Initialize Variable valuestreamerTaskId	Parameter <table><tr><th>Key</th><th>Value</th></tr><tr><td>Name:</td><td>valuestreamerTaskId</td></tr><tr><td>Type:</td><td>String</td></tr></table>	Key	Value	Name:	valuestreamerTaskId	Type:	String						
Key	Value														
Name:	valuestreamerTaskId														
Type:	String														
4.	For-Each-Loop to iterate over all returned rows from step 2. Power Automate does not know that there always will be only one returned row in this usecase.	For each: Control, For each	Iterate over @outputs('List_rows_of_task_table,_based_on_planner_task_ID')?['body/value']"												
5.	Set variable valuestreamerTaskID to UUID from task table of the current loop.	Set variable: Variables, Set variable valuestreamerTaskId to UUID from table	Parameter <table><tr><th>Key</th><th>Value</th></tr><tr><td>Name:</td><td>valuestramerTaskID</td></tr><tr><td>Value:</td><td>@items('For_each')?['cr4b3_uuid']</td></tr></table>	Key	Value	Name:	valuestramerTaskID	Value:	@items('For_each')?['cr4b3_uuid']						
Key	Value														
Name:	valuestramerTaskID														
Value:	@items('For_each')?['cr4b3_uuid']														
6.	Send HTTP-Request (GET) to Valuestreamer, filtered by the value of valuestreamerTaskId, to get the data of the task in the ValueStreamer.	HTTP: HTTP, HTTP - GET task infos	Parameter <table><tr><th>Key</th><th>Value</th></tr><tr><td>Uri:</td><td>YourDomain/api/exchange/tasks/variables('val uestreamerTaskID')</td></tr><tr><td>Method:</td><td>Get</td></tr><tr><td>Headers:</td><td></td></tr><tr><td>Authorization</td><td>Basic YOURTOKEN</td></tr><tr><td>Accept</td><td>application/json</td></tr></table>	Key	Value	Uri:	YourDomain/api/exchange/tasks/variables('val uestreamerTaskID')	Method:	Get	Headers:		Authorization	Basic YOURTOKEN	Accept	application/json
Key	Value														
Uri:	YourDomain/api/exchange/tasks/variables('val uestreamerTaskID')														
Method:	Get														
Headers:															
Authorization	Basic YOURTOKEN														
Accept	application/json														

			Queries: includeAll teamYourTeam-ID progress CREATED,PLANNED,INPROGRESS
7.	Parse the response of step 6 in a JSON-Object.	Parse JSON: Data Operations, Parse JSON from GET Task	Parameter Content: body('HTTP_reqeust_to_GET_all_tasks_from_VS_Team') JSON-Schema { "type": "object", "properties": { "id": { "type": "string" }, "actualEffort": { "type": "integer" }, "actualEndDate": {}, "actualStartDate": {}, "archivedDate": {}, "category": { "type": "string" }, "createdBy": { "type": "string" }, "createdByTeam": { "type": "string" }, "createdDate": { "type": "string" }, "label": {}, "linkedToProcess": {}, "linkedToProcessitem": {}, "linkedToTeam": {}, "modifiedBy": {}, }

			<pre>"modifiedDate": {}, "priority": { "type": "boolean" }, "progress": { "type": "string" }, "responsiblePerson": {}, "responsibleTeam": { "type": "string" }, "status": { "type": "integer" }, "statusMessage": {}, "targetEffort": { "type": "number" }, "targetEndDate": { "type": "string" }, "targetStartDate": {}, "title": { "type": "string" }, "tCard": { "type": "boolean" } } } }</pre>																
8.	Send HTTP-Request (Put) to ValueStreamer to set the task in ValueStreamer to done.	HTTP: HTTP, HTTP - PUT task to done	▼ Parameter <table><tr><th>Key</th><th>Value</th></tr><tr><td>Uri:</td><td>/api/exchange/tasks/variables('valuestreamerTaskID')</td></tr><tr><td>Method:</td><td>PUT</td></tr><tr><td colspan="2">Headers</td></tr><tr><td>Authorization</td><td>Basic YOURTOKEN</td></tr><tr><td>Accept</td><td>application/json</td></tr><tr><td colspan="2">Body:</td></tr><tr><td>1</td><td>{</td></tr></table>	Key	Value	Uri:	/api/exchange/tasks/variables('valuestreamerTaskID')	Method:	PUT	Headers		Authorization	Basic YOURTOKEN	Accept	application/json	Body:		1	{
Key	Value																		
Uri:	/api/exchange/tasks/variables('valuestreamerTaskID')																		
Method:	PUT																		
Headers																			
Authorization	Basic YOURTOKEN																		
Accept	application/json																		
Body:																			
1	{																		

			2"progress": "DONE", 3"targetEndDate": "@{body('Parse_JSON_from 4"targetEffort": "@{body('Parse_JSON_from_G 5"category": "@{body('Parse_JSON_from_GET_ 6"priority": "@{body('Parse_JSON_from_GET_T 7"responsibleTeam": "@{body('Parse_JSON_fr 8"responsiblePerson": "@{body('Parse_JSON_ 9"targetStartDate": "@{body('Parse_JSON_fr 10"tCard": "@{body('Parse_JSON_from_GET_Tas 11"label": "@{body('Parse_JSON_from_GET_Tas 12"actualEffort": "@{body('Parse_JSON_from_ 13"status": "@{body('Parse_JSON_from_GET_Tas 14"title": "@{body('Parse_JSON_from_GET_Tas 15}						
9.	For-Each-Loop to iterate over all returned rows from step 2. Power Automate does not know that there always will be only one returned row in this usecase.	For each: Control, For each	Iterate over "@outputs('List_rows_of_task_ table,_based_on_planner_task_ ID')?['body/value']"						
10.	Update rows in task table by setting progress to done.	Update a row: Micorosoft Dataverse, Update a row	Parameter <table><tr><th>Key</th><th>Value</th></tr><tr><td>Table Name:</td><td>YourTaskTable</td></tr><tr><td>Row_ID:</td><td>"@items('For_each_1')?</td></tr></table>	Key	Value	Table Name:	YourTaskTable	Row_ID:	"@items('For_each_1')?
Key	Value								
Table Name:	YourTaskTable								
Row_ID:	"@items('For_each_1')?								

['cr4b3_vs_ejotpoc_tasksid']

"

Progress: DONE

| 11. | Check if the task is a tcard in the ValueStreamer. If the card is a tcard, then the condition is true. Otherwise the condition is false. If true: Go to step 11.1.1 If false: There is nothing to do - end flow. | **Condition:** Control, Condition Check if task is tcard | Condition body('Parse_JSON_fromGet_Task')?['tCard'] is equal to true |

11.1.1	Initialize new variable with Name newDueDate of Datatype String and a value. This variable will be used to create a new task in valuestreamer (Step 11.1.2). Be aware that there might be errors if the value is earlier than today. Depending on your usecase the value can be different. Usecase 1: The targetEndDate of the new task is one week later from today. Usecase 2: The targetEndDate of the new task is one week later than the old targetEndDate.	Initialize Variable: Variables, Initialize variable newDueDate and set value	Parameter <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Name:</td><td>newDueDate</td></tr><tr><td>Type:</td><td>String</td></tr><tr><td>Value: (Usecase 1)</td><td>getFutureTime(7, 'Day', 'yyyy-MM-dd')</td></tr><tr><td>Value: (Usecase 2)</td><td>addDays(body('Parse_JSON_from_GET_Task')['targetEndDate'], 7, 'yyyy-MM-dd')</td></tr></tbody></table>	Key	Value	Name:	newDueDate	Type:	String	Value: (Usecase 1)	getFutureTime(7, 'Day', 'yyyy-MM-dd')	Value: (Usecase 2)	addDays(body('Parse_JSON_from_GET_Task')['targetEndDate'], 7, 'yyyy-MM-dd')
Key	Value												
Name:	newDueDate												
Type:	String												
Value: (Usecase 1)	getFutureTime(7, 'Day', 'yyyy-MM-dd')												
Value: (Usecase 2)	addDays(body('Parse_JSON_from_GET_Task')['targetEndDate'], 7, 'yyyy-MM-dd')												
11.1.2	The completed task is a tcard. Therefore an identical task has to be created in the ValueStream. Send a HTTP-Request (Post) to ValueStream to create an identical task with a value for “targetEndDate” using the variable newDueDate from step 11.1.1. End of flow.	HTTP: HTTP, HTTP - Post a new task in ValueStream	Parameter <table><thead><tr><th>Key</th><th>Value</th></tr></thead><tbody><tr><td>Uri:</td><td>/api/exchange/tasks</td></tr><tr><td>Method:</td><td>Post</td></tr></tbody></table> Headers <table><tbody><tr><td>Authorization</td><td>Basic YOURTOKEN</td></tr><tr><td>Accept</td><td>application/json</td></tr></tbody></table> Body: <pre>1 { 2 "targetEndDate": "@{variables('newDueDate')}" 3 "targetEffort": @{{body('Parse_JSON_from_GET_Task')['targetEffort']} 4 "category": @{{body('Parse_JSON_from_GET_Task')['category']} 5 "priority": @{{body('Parse_JSON_from_GET_Task')['priority']} 6 "responsibleTeam": @{{body('Parse_JSON_from_GET_Task')['responsibleTeam']} 7 "responsiblePerson": @{{body('Parse_JSON_from_GET_Task')['responsiblePerson']} 8 "tCard": @{{body('Parse_JSON_from_GET_Task')['tCard']} 9 "label": @{{body('Parse_JSON_from_GET_Task')['label']} 10 "status": @{{body('Parse_JSON_from_GET_Task')['status']} 11 "title": @{{body('Parse_JSON_from_GET_Task')['title']} 12 }</pre>	Key	Value	Uri:	/api/exchange/tasks	Method:	Post	Authorization	Basic YOURTOKEN	Accept	application/json
Key	Value												
Uri:	/api/exchange/tasks												
Method:	Post												
Authorization	Basic YOURTOKEN												
Accept	application/json												

Steps as screenshots

Following table shows some Screenshots which belong to a certain step from above.

Step	Screenshot
------	------------

1.

[←](#) Update ValueStream task when Planner task is completed

When a task is completed

**Parameters** Settings Code View About

Group Id *

YourGroupId



Plan Id *

Test PoC



▼ How often do you want to check for items?

Recurrence *

Interval *

1

Frequency *

Minute



Time Zone

Select timezone.



Start Time

Example: 2017-03-24T15:00:00Z

2.

[←](#) Update ValueStream task when Planner task is completed

List rows of task table, based on planner task ID

**Parameters** Settings Code View Testing About

Table Name *

VS_EJOT-PoC_Tasks



Advanced parameters

Showing 1 of 8



Show all

Clear all

Filter Rows

cr4b3_plannerid eq 'Id ×'



Id ×'



Enter an OData style filter expression to limit which rows are listed

3.

[←](#) Update ValueStream task when Planner task is completed

Initialize variable valuestreamerTaskId

**Parameters** Settings Code View About

Name *

valuestreamerTaskId

Type *

String



Value

Enter initial value

4.

← Update ValueStreamer task when Planner task is completed

For each

Parameters Settings Code View About

Select An Output From Previous Steps *

body/value ×

outputs('List_rows_of_task_table_based_on_planner_task_ID')['body/value']

5.

← Update ValueStreamer task when Planner task is completed

Set variable valuestreamerTaskId to UUID from table

Parameters Settings Code View About

Name *

valuestreamerTaskId

Value *

cr4b3_uuid ×

items('For_each')['cr4b3_uuid']

6.

HTTP - GET task infos

Parameters Settings Code View About

URI *

YourVSDomain /api/exchange/tasks/ {v} valuestre... ×

Method *

GET

Headers

Authorization	Basic YourToken
Accept	application/json

Queries

Enter key	Enter value
-----------	-------------

Body

variables('valuestreamerTaskId')

7.

← Update ValueStreamer task when Planner task is completed



Parse JSON from GET Task



Parameters Settings Code View About

Content *



Body *



Schema *

body('HTTP_-GET_task_infos')

```
{
  "type": "object",
  "properties": {
    "id": {
      "type": "string"
    },
    "actualEffort": {
      "type": "integer"
    },
    "actualEndDate": {},
    "actualStartDate": {},
    "archivedDate": {},
    "category": {
      "type": "string"
    },
    "createdBy": {
      "type": "string"
    },
    "createdByTeam": {
      "type": "string"
    }
  }
}
```

[See sample payload to generate schema](#)

8.

← Update ValueStreamer task when Planner task is completed

URI *

https:// **YourDomain** /api/exchange/tasks/ **valuestre...**



Method *

PUT

variables('valuestreamerTaskId')

Headers

Authorization	Basic YourToken
Accept	application/json

Queries

Enter key	Enter value
-----------	-------------

Body

```
{
  "progress": "DONE",
  "targetEndDate": "Body tar... ",
  "targetEffort": "Body tar... ",
  "category": "Body cat... ",
  "priority": "Body prio... ",
  "responsibleTeam": "Body res... ",
  "responsiblePerson": "Body res... ",
  "targetStartDate": "Body tar... ",
  "tCard": "Body tCard ",
  "label": "Body label "
}
```

9.

← Update ValueStreamr task when Planner task is completed

 For each 1 ⋮ <

Parameters Settings Code View About

Select An Output From Previous Steps *

 body/value × 

`outputs('List_rows_of_task_table_based_on_planner_task_ID')?['body/value']`

10.

← Update ValueStreamr task when Planner task is completed

 Update a row ⋮ <

Parameters Settings Code View Testing About

Table Name *

VS_EJOT-PoC_Tasks ▼

Row ID *

 cr4b3_vs... × 

UUID

`items('For_each_1')?['cr4b3_vs_ejotpoc_tasksid']`

Advanced parameters

Showing 1 of 13 ▼ Show all Clear all

Progress

DONE ×

11.

← Update ValueStreamr task when Planner task is completed

 Condition Check if task is tcard ⋮ <

Parameters Settings Code View About

Condition Expression *

Provide the values to compare and select the operator to use.

AND ▼

 Body tCard × is equal to ▼  true × ...

+ New item ▼

`body('Parse_JSON_from_GET_Task')?['tCard']`

11.1.1 (Usecase 2)

← Update ValueStreamer task when Planner task is completed

 Initialize variable newDueDate and set value

Parameters

Settings

Code View

About

Name *

Type *

Value

 addDays(...) ×



addDays(body('Parse_JSON_from_GET_Task')['targetEndDate'],7,'yyyy-MM-dd')

11.1.2

URI *

 YourDomain /api/exchange/tasks

Method *

Headers

Authorization	Basic YourToken	×	
Accept	application/json		

Queries

Enter key	Enter value	
-----------	-------------	---

Body

```
{
  "targetEndDate": " {x} newDue... × ",
  "targetEffort": {v} Body tar... ×
  "category": " {v} Body cat... × " variables('newDueDate')
  "priority": {v} Body prio... ×
  "responsibleTeam": " {v} Body res... × ",
  "responsiblePerson": " {v} Body res... × ",
  "tCard": " {v} Body tCard × ",
  "label": " {v} Body label × ",
  "status": " {v} Body stat... × ",
  "title": " {v} Body title × "
}
```

