Excel GroupBy Example Phase I

The document outlines the entire process, requirements, build-up and exact expressions to support the flow and solution.

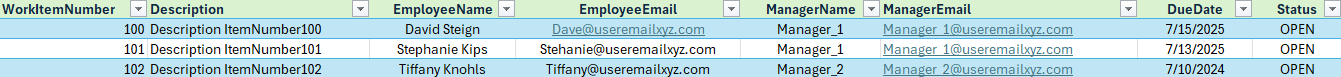
# Business Need

Every night an Excel report is generated that maps all work items, based on Organization, Manager and Employee.

The report contains the following details:

* **ID**: the work item number
* **Description**: a short description of the work item
* **EmployeeName**: the employees display name
* **EmployeeEmail**: the employees email address
* **ManagerName**: the employees managers display name
* **ManagerEmail**: the employees managers email address
* **DueDate**: the work item due date
* **Status**: the work item status (OPEN, CLOSED, NEW)
* **Org**: the organization that the manager and employee are in

And it looks like this:



Below are the requirements for Phase 1

# Requirements

Below are the requirements that we gathered from the business. It also contains additional requirements that we want to follow based on best practices and IT support.

## Business Requirements

1. An Excel file will be received via Email every day that provides work item details and must be processed immediately
2. The daily file will replace the previous file completely, but we must store each days file for compliance
3. The Subject of the email will be WorkItems and the Attachment count will always be 1, with a file called workitems.xlsx
4. The incoming Excel file should be stored in the GroupBy/WorkItems folder, in the document library Excel-Samples, in the SharePoint site, TheGernaeyCode
5. The incoming file, when saved, should insert the date and time formatted as \_dd\_MM\_yyyy\_\_hh\_mm\_ss\_fffz to the file name.
6. **Note: fffz are utc milliseconds**
7. **Example**:
8. **Original Name**: workitems.xlsx
9. **Processed Named**: workitems\_10\_03\_2025\_\_12\_24\_43\_333Z.xlsx
10. Extract the data from Excel and based on the scenario, send a properly formatted email to the appropriate recipients
11. Once processed the Excel file should be moved to the GroupBy/Processed folder
12. Any failed documents should be moved to the GroupBy/Failed folder
13. For each file, regardless of processing outcome, an email should be sent to the [workitemsgroup@useremailxyz.com](mailto:workitemsgroup@useremailxyz.com) with the file name, a link to the file in SharePoint and its processing status

## IT Requirements

1. Enable notifications for processing
2. Enable logging of Flow state within the Details page of the Power Automate flow
3. When notifying, specify whether it is a failed flow run or a successful flow run
4. When a failure occurs send the details within the email body
5. Provide error handling within the flow

# Building the Flow Logic

This section covers the functional overview of what is being asked. It will cover the following:

1. The functional overview of steps based on requirements
2. A Visio diagram that covers the flow
3. In the next section will be the Technical specifications

## Functional Overview

This area covers the functional details that will be implemented.

Based on our requirements here are the functional steps we need to take:

1. Schedule our flow to run anytime we receive an email with the following details
   * Has attachments
   * Has only 1 attachment
   * The subject is workitems
   * The attachment name is workitems.xlsx  
     + If these conditions are not met, then we need to send an email stating that we had an issue with email processing and we need to log in the flows Details run section that it failed
     + If these conditions are true, then the next thing we must do is define our new file name to match the required format
2. Next, we need to create our new file in SharePoint, based on the locations mentioned in the requirements and given the appropriate name format
3. Now that we have our file, the requirement is to look through the rows of data and generate a report for our scenarios
   * A report email goes to each Employee, with their items only and CC the manager
   * An employee rollup report email goes to each Manager, with all their employees’ items
4. Depending on the status of our processing of the Email (Excel file), we need to
   * Move the file to either the Processed folder, if processed successfully, or to the Failed folder
   * Send a status email to the operations team, named [workitemsgroup@useremailxyz.com](mailto:workitemsgroup@useremailxyz.com)
5. In addition to these business requirements, we must
   * Send a status message to [workitemsgroup@useremailxyz.com](mailto:workitemsgroup@useremailxyz.com) based on success or failures
   * Log in the Power Automate Flows Details the completion status of the Flow, which will be either Failed or Success

## Visio Diagram of our Business Process

The below is a Visio that is based on the business requirements, turned into a functional flow, which can then be translated directly into a Technical Specification, and directly into our technical implementation.



# Technical Specification

After reviewing the business requirements and the business process functional flow, the below technical implementation details define how we will solve the business problem, while ensuring that we have operational visibility into the flow’s actions.

These are the step by step instructions for the flow broken down by its section

## Flow Trigger

We will leverage the “When a new email arrives (V3)” trigger and will configure it as such.

* Folder: Inbox
* Include Attachments: Yes
* Subject Filter: WorkItems
* Only with Attachments: Yes
* The rest leave blank or their defaults  
  A screenshot of a computer screen

  AI-generated content may be incorrect.

## Email Details Verification

We will leverage a Condition action as seen below to verify if all the conditions are met to process this email

A screenshot of a computer

AI-generated content may be incorrect.

It will have the following configuration: 4 checks, all “Anded” together. This means they must all be true to continue.

* **Validate that the length of the Attachments is equal to 1**
  + We do that by checking that the length of the array of attachments is equal to 1. Configure the condition boxes from left to right as such:
    - length(triggerOutputs()?['body/attachments'])
      * This is the expression length, with the Dynamic trigger property
    - is equal to
      * simply select this in the drop down
    - 1
      * just type in the #1
* **Validate the name of the first attachment**. Remember the attachments are an array of data. We only expect 1 item, so we always check the arrays [0] item index
  + From left to right in the boxes
    - triggerOutputs()?['body/attachments']?[0]?['name']
      * This is a fully typed expression that grabs the name of the first attachment
    - is equal to
      * simply select this in the drop down
    - workitems.xlsx
      * This is the hardcoded name of the file expected to be attached. In real-world examples we would not hard code it in case it changes
* **Validate the subject is WorkItems by intentionally converting the value to all lower case and then comparing it to the intended value workitems**
  + From left to right in the boxes
    - toLower(triggerOutputs()?['body/subject'])
      * use the toLower expression to convert the Dynamic property Subject to all lower case
    - is equal to
      * again select this
    - type workitems, in the Dynamic content window and click ok/update
* **Validate that the Email even has attachments**
  + From left to right
    - The dynamic property Has Attachments from the Trigger
      * Click in the Dynamic Property and scroll down until you find the Has Attachments property of the Trigger. You may need to click Show More in the window to see it
    - Is equal to
      * Select from drop down
    - True 🡸 the expression
      * Click the expression tab type true, select the true expression when it appears. If you just type the words true, it will not be in a PINK box and will be wrong

Once the condition has determined whether or not all the checks are true (if yes) or false (if no), we will perform the following:

### The If No Side

If the details have even 1 false statement it will go into the if no. Here we will leverage 2 conditions:

A screenshot of a computer

AI-generated content may be incorrect.

1. Send an email (V2) action from Office 365, which will allow us to configure it to send an email to operations to tell them that the flow failed
   1. This was simply configured with a hardcoded email address, message and subject  
      A screenshot of a computer

      AI-generated content may be incorrect.
2. The Terminate action, allows us to tell the flow to stop and to log a specific message and status.
   1. Status: Failed 🡸 select from drop down
   2. Code: 500 🡸 I made this number up
   3. Message: Attachment, Subject, or Has Attachments Failed 🡸 I also made this up, you can type anything or use dynamic values or variables

A screenshot of a computer

AI-generated content may be incorrect.

## The If Yes Side

If all conditions were met, then we will move onto our primary processing steps. Below are those steps and their configuration. The image below represents all the actions and steps, but I will break it down below the image.

A screenshot of a computer

AI-generated content may be incorrect.

### Try, Catch, Success Blocks (aka Scopes)

In Power Platform you can implement a Try Catch error handling routine with Scopes and the proper configuration of the Run After properties of each Scope.

A screenshot of a computer

AI-generated content may be incorrect.

In this case we needed 3 scopes:

1. The Try Scope: holds all our primary actions. We would place all the actions that we want to capture errors for here.   
   Note: it’s possible to have multiple hierarchical try catches, especially in cases where you might be looping through data and do not want the entire flow to fail because a loop does but we will not be using that in our flow today
2. The Catch Scope: is used to capture if any action in the Try scope fails.
   1. Configuration:
      1. Click the … in the header and select Configure run after and change per the image below
      2. A screenshot of a computer

         AI-generated content may be incorrect.
   2. In the Send Email V2 below configure as such
      1. Put the email address you want success and failure notifications to go to. In my case I added a fake email address as a demonstration. But to really receive it change it to a real email
      2. For the subject, you can see my text and then –
         * After the dash is the expression   
           formatDateTime(utcNow(), ‘dd/MM/yyyy’)
      3. In the body, I have an expression result(‘Try’). This expression will send me information related to what happened in the Try scope. The result expression itself allows you to grab the “result” of any action as a description or summary if you will
   3. For the Terminate action which I renamed Failed for cause, I selected the Failed option and then added a generic 501 code and message that I made up myself.
   4. Below is an image of both actions within the Catch scope block
   5. A screenshot of a computer

      AI-generated content may be incorrect.
3. The Success Scope: This should be used when the Try Scope succeeds.  
   NOTE: The default configuration for this is Success, so we can leave it as configured
   1. In the Success Scope, you do NOT have to change the Configure run after, as the default of any action is Success
      1. Inside we will setup the Email the same as in the Catch scope, with the same expressions, but with a different body since it worked we do not need to send a summary of what happened in the Try scope
      2. As for the terminate action, this time we selected Succeeded and that’s it
   2. A screenshot of a computer

      AI-generated content may be incorrect.

# Summary

In this example we have demonstrated the following:

* How to configure an automated flow to trigger when an Email is received and to process the attachment
* How to save the attachment to SharePoint document library
* How to process the Excel file by listing all the rows
* How to get our unique Email Addresses from the list of data
* How to email each unique employee only their work items
* To notify operations of the flows processing status and log success or failure in Flow